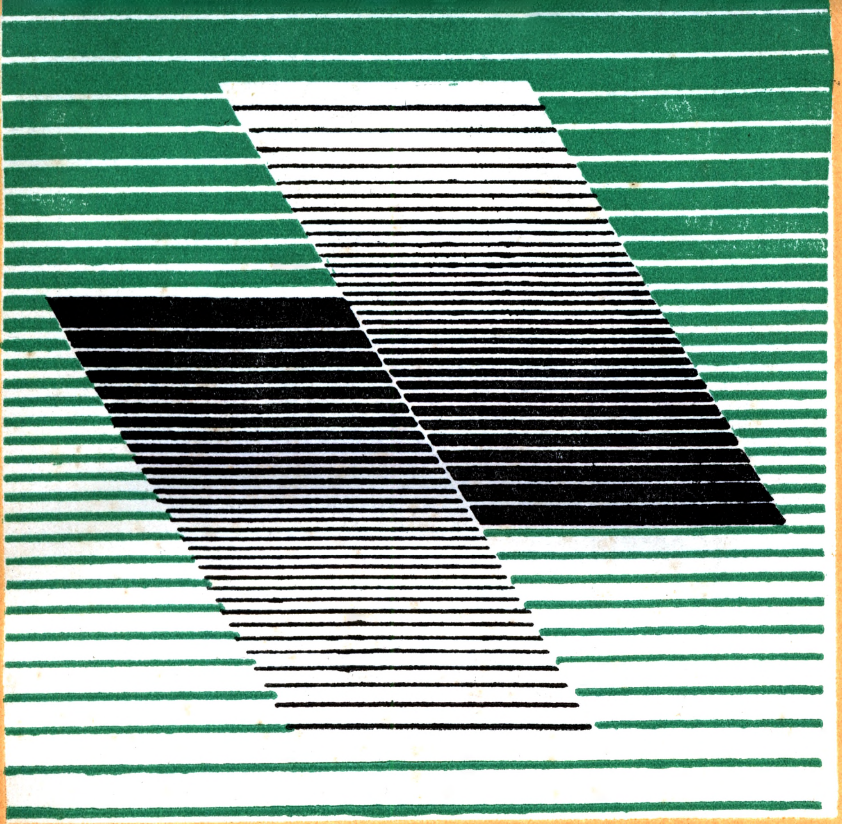


CEC—I LOGO编程百例

杨周武 编

陕西师范大学出版社

CEC-I 中华学习机

LOGO 编程百例

杨周武　编著

0517 8834

深圳圖書館藏書
A
COLLECTION
SHENZHEN LIBRARY

陕西师范大学出版社

（陕）新登字 008 号

内 容 提 要

本书以CEC-I型中华学习机为对象，也适用于APPLEⅡ及其兼容机的MIT LOGO版本。

书中汇集常见题目110个。对每个题目的程序设计都作了算法分析，编程思路、方法与技巧的仔细说明。内容丰富充实，方法灵活多样，语言文字通顺简练。每个程序都有运行示例或结果。

本书是提高LOGO语言程序设计能力和技巧的有益读物。适合于广大青少年，大、中、小学学生，初学计算机的人及家庭使用。对于中小学教师及科技工作者，也有参考价值。

CEC-I中华学习机

LOGO编程百例

杨周武 编著

*

陕西师范大学出版社出版发行

（西安市陕西师大120信箱）

陕西省新华书店经销　西安电子科技大学印刷厂印刷

*

开本 787×1092　1/32　印张 9.75　字数 210千

1992年2月第1版　1992年2月第1次印刷

印数：1—5000

ISBN 7-5613-0487-1

G·357　定价 4.50 元

前　　言

随着当代科学技术的迅速发展，电子计算机及其应用，日益广泛。对于中小学生及广大青少年，普及计算机教育，也日益重视。

国家教委要求在全国推广普及LOGO语言。电子部、清华大学、陕西计算机厂等联合设计生产了CEC-I型中华学习机，机内固化了LOGO语言系统，这就为普及LOGO语言奠定了一定的物质基础。加上LOGO语言本身结构良好，使用方便，功能较强，独具特色和魅力，对于开发青少年智力，提高能力，有其独到之处。可以想见，LOGO语言今后将会成为中小学生、广大青少年，甚至成人，学习计算机的基本语言之一。

针对目前已经开展LOGO语言教学的学校、校外辅导站及青少年活动中心，感到LOGO语言方面的资料较少，题目较缺的现状，将几年来在LOGO语言教学方面的材料，以及在报刊上发表过的有关文章，收集整理和充实，汇集出110个题目，从程序设计的算法分析、编程思路、编程方法与技巧方面，都尽量给予详细必要的分析与说明。罗列出简明的程序，并给出运行结果或示例。较多题目，从不同角度给出不同的编程方法。这对于提高读者程序设计能力和技巧，定会有所帮助。

本书的出版，曾得到孟庆玉、刘子芳老师及王兵同学的大力支持和帮助。书末收录了王兵的两个软件。

同时还要感谢陕西师范大学出版社的大力协助。

书中难免疏漏和错误，恳切欢迎读者批评指正。

为了方便读者，本书中全部程序及软件，配有一套软磁盘片。需要者可向出版社或作者直接联系。

编　者

1990年10月于西安

目　　录

一　简单图案与绘图……………………………………（ 1 ）

1-1 打印数字三角①　1-2 打印数字三角②　1-3 打印字符菱形图案　1-4 打印空心菱形图案　1-5 打印数字菱形　1-6 打印夹行菱形　1-7 画正方形套　1-8 画“三菱”图形　1-9画伞　1-10 画灯笼　1-11 画圆　1-12 画椭圆　1-13 画三叉树

二　常用过程……………………………………………（ 27 ）

2-1 字函数过程(7 条)　2-2 表函数过程(8 条)　2-3 计算常用的函数过程(15 条)　2-4 FOR 循环过程　2-5 一维数组　2-6 二维数组

三　数值计算……………………………………………（ 48 ）

3-1 求自然数的和　3-2 求若干个加数的和　3-3 求 T＝N!　3-4 父子年龄问题　3-5 求 S＝1!＋2!＋…＋10! 的值　3-6求分式值①　3-7 求分式值②　3-8 求 N 个数的最大公约数及其最小公倍数　3-9 人口问题　3-10 解一元二次方程　3-11 母牛生殖问题　3-12 小兔繁殖问题　3-13 求一个数列的前N项　3-14 求式中字母代表的值①　3-15 求式中字母代表的值②

四　有趣的数……………………………………………（78）
4-1 分 316 为二正数之和　4-2 求 N! 能被M的最高多少次方整除　4-3 尾巴为6的数　4-4 平方中含 4 的数　4-5“水仙花”数　4-6 和位幂数　4-7 同构数　4-8 回文数　4-9 有趣的整数　4-10 不怕撕裂的数　4-11 求质数　4-12 分解质因数　4-13 史密斯数　4-14 卡布列克运算
五　古算趣题……………………………………………（112）
5-1 鸡兔同笼问题　5-2 百马百瓦问题　5-3 角谷猜想　5-4梵塔问题　5-5棋盘奖赏　5-6 约瑟夫斯问题　5-7 发扑克牌问题　5-8 奇妙的素数算式　5-9 杨辉三角形　5-10 孙子定理
六　高精度计算……………………………………………（141）
6-1 高精度加法　6-2 高精度减法　6-3 高精度乘法　6-4 高精度阶乘　6-5 高精度乘方　6-6 高精度除法
七　数制转换……………………………………………（153）
7-1 10 进制数转换成 2、8、16 进制数　7-2 0～255 间 10 进制数对应转换成 2、8、16 进制数　7-3 任意两种进制间数的转换
八　比较与排序……………………………………………（161）
8-1 打印最大的一个数　8-2 同时打印最大和最小的一个数　8-3 数字排序　8-4 字母按 ASCII 码排序　8-5 随机字母组的字典

排序 8-6 英文人名字典排序 8-7 映射排序应用
九 统计应用……………………………………………（176）
9-1 选票统计① 9-2 选票统计② 9-3 成绩统计 9-4 及格人数统计 9-5 分数分段统计
十 组合问题……………………………………………（187）
10-1 由1～4组成的三位数 10-2 由0～9组成的三位数 10-3 无零三位数和等于13的数 10-4 用一元、二元、五元凑十元的方法 10-5 由a、b、c组成n位数的种数
十一 逻辑推理问题……………………………………（198）
11-1 好人好事是谁做的？ 11-2 对竞赛名次的预测 11-3 湖泊大小次序问题 11-4 破案
十二 近似计算应用……………………………………（210）
12-1 求n次多项式的值 12-2 二分法求方程近似实根 12-3 优选法求方程近似实根 12-4 迭代法求方程近似实根 12-5 梯形法求定积分的值 12-6 优选法求函数极大点
十三 杂题……………………………………………（225）
13-1 改变矩阵形式 13-2 把一个数分成二位、三位、四位数 13-3 圆盘上找数 13-4 色彩演示 13-5 按要求打印英文单词 13-6 掷硬币 13-7 掷骰子 13-8 豆

子下落实验 13-9 字符组成移动的飞机
13-10 打印九九乘法表 13-11 小学生除
法演算练习程序 13-12 念数程序 13-13
判断闰年 13-14 查星期程序 13-15 打
印年历程序 13-16 趣味清屏程序
附录 软件制作………………………………………（274）
一《LOGO 键控绘图》软件简介
二《LOGO 音乐制作》软件简介

参考书目及文献…………………………………………（304）

一　简单图案与绘图

1-1　打印数字三角①

编程打印如图 1-1 所示的数字三角。

```
        1
       121
      12321
     1234321
    123454321
   12345654321
  1234567654321
 123456787654321
12345678987654321
```

图　1-1

解　行数不断增大，每一行上前半由小到大，后半又由大到小，并与行数有关。

我们编制四个过程，其中 Z 为主过程，控制程序的执行。其余 A、B、C 为子过程。过程 A 确定总行数及每一行起始打印位置。B 与 C 形式相似，但打印命令与递归位置不同，使每行上的数字由小到大又由大到小。这对于理解递归的执行过程是有益的。

程序如下(只要键入 Z，回车，即得图 1-1)：

程序 1-1

```
TO Z
  CLEARTEXT
  A 1
END

TO A :N
  IF :N > 9 STOP
  CURSOR 21 - :N :N
  B :N
  C :N - 1
  PR "
  A :N + 1
END

TO B :N
  IF :N = 0 STOP
  B :N - 1
  PRINT1 :N
END

TO C :N
  IF :N = 0 STOP
  PRINT1 :N
  C :N - 1
END
```

练习：如何编程打印出如图 1-2 所示的数字三角？

```
        1
       212
      32123
     4321234
    543212345
   65432123456
  7654321234567
 876543212345678
98765432123456789
```

图 1-2

1-2 打印数字三角②

编程打印如图 1-3 所示的数字三角。

```
        1
       222
      33333
     4444444
    555555555
   66666666666
  7777777777777
 888888888888888
99999999999999999
```

图 1-3

解法1 用一个重复命令的二重嵌套过程，使外层确定行数，内层确定列数，问题即可解决。共 9 行，每行上2I-1个数。为了居屏幕中央，从 I+5 即第六行开始，每行上左起第一个数字在 21-I 列上打印。过程为 ZSJ。

程序如下：

程序 1-2-1

```
TO ZSJ
```

```
  CLEARTEXT MAKE "I 1
  REPEAT 9 [CURSOR 21 - :I :I+5 REPEA
T :I*2-1 [PRINT1 :I] PR "" MAKE "I
:I+1]
END
```

?ZSJ 回车，即可得图 1-3。

解法 2 建立两个递归过程，一个控制行数及每一行上的起始位置。另一个控制列数，即每一行上的数字及数字的个数。主过程 SJ1，子过程 SJ2，运行可得同样结果。

程序 1-2-2

```
TO SJ1 :I
  IF :I > 9 STOP
  CURSOR 21 - :I :I + 5
  SJ2 1
  SJ1 :I + 1
END

TO SJ2 :J
  IF :J > :I * 2 - 1 PR " STOP
  PRINT1 :I
  SJ2 :J + 1
END
```

练习：能否只修改上述主过程中的一个地方，而使 I 从 1 开始，运行 SJ1，打印出如图 1-4 所示的图案。即

?SJ1 1

```
        1
       22
      333
     4444
    55555
   666666
  7777777
 88888888
999999999
```

图 1-4

1-3 打印字符菱形图案

试编程打印如图 1-5 所示的字符菱形图案。要求菱形的行数随不同的输入而发生变化。

```
  *
 ***
*****
 ***
  *
```

图 1-5

解 打印菱形字符图案，方法多式多样。譬如分为上、下两部分，各由一个子过程完成。也可以上下统一考虑，增加一个求绝对值的函数过程，而由一个过程全部打出。无论如何考虑，关键是必须解决三个问题。一是每行字符的起始位置。其次是每行打印字符的个数。最后是打印时的终止条件。

法 1 将菱形分为上下两部分。分别由子过程 LXS 和 LXX 各自完成一部分。主过程 LX 总控，它带一个输入 M，

给出菱形的行数。

程序及运行示例如下。

程序 1-3-1

```
TO LX :M
  CLEARTEXT
  LXS 1 :M
  LXX ( :M + 1 ) / 2 + 1 :M
END

TO LXS :N :H
  CURSOR 21 - :N :N
  IF :N > ( :H + 1 ) / 2 STOP
  REPEAT :N * 2 -1 [PRINT1 "*]
  PR ""
  LXS :N + 1 :H
END

TO LXX :N :H
  IF :N > :H STOP
  MAKE " A 21 - ( :M - :N + 1 )
  MAKE " B 2 * :M - :N * 2 + 1
  CURSOR :A :N
  REPEAT :B [PRINT1 "*]
  PR ""
  LXX :N + 1 :H
END
```

?LX 11 回车，即可得共有 11 行的菱形。

法 2 菱形上下以(N+1)/2 行为对称，各行的字符个数及起始位置都是相同的。这样我们可以通过绝对值函数过程来确定各对称行字符的起始位置，即它所在的列。见主过程 LX 第二行光标位置命令后的列位置式。而各行字符的个数，对于对称行来讲，也是相同的。它可由子过程 PL 第一行条件式的后半段确定。这样，不难写出如下的打印菱形字符图案的程序。

程序 1-3-2

```
TO LX :I :N ; 1-N
  IF :I > :N STOP
  CURSOR ( ABS ( :N + 1 ) / 2 - :I)
+ 1 :I
  PL 1
  LX :I +1 :N
END

TO PL :J ; 1--N-2*B
  IF :J > :N - 2 * ( ABS ( :N + 1 )
/ 2 - :I ) PR "'' STOP
  PRINT1 "*
  PL :J + 1
END

TO ABS :X
  IF :X < 0 OP - :X
  OP :X
```

END

?LX 1 9 回车，则能得到一个有 9 行的菱形。

练习：编程打印出如图 1-6 所示"三菱"形字符图案。

```
          *
         ***
        *****
       *******
        *****
         ***
          *
     *****  *****
    *****    *****
   *****      *****
  *****        *****
```

图 1-6

1-4 打印空心菱形图案

打印如图 1-7 所示的空心菱形。要求根据输入的行数，可以打印出任意大小的菱形。

```
    *
   * *
  *   *
 *     *
*       *
 *     *
  *   *
   * *
    *
```

图 1-7

解 关键是如何确定每一行上星号的位置。建立一个求绝对值的函数过程 ABS，这样就可以把菱形上半部与下半部统一起来。菱形总行数若为 N，使行数 I 从 1 到N，将(N+1)/2－I 的绝对值赋给变量 LS，则在 I 行的第一个星号应在 LS+1 列，第二个星号应在(N－LS)列。

主过程 KLX，带两个输入。前一个由 1 开始，后一个为总行数。

程序及运行示例如下：

程序 1-4

```
TO KLX :I :N ; 1-N
  IF :I > :N STOP
  MAKE "LS ( ABS ( :N + 1 ) /2 - :I )
  IF ANYOF :I = 1 :I = :N CURSOR :L
S + 1 :I PR "* ELSE CURSOR :LS + 1
:I PRINT1 "* CURSOR :N - :LS :I PR "*
  KLX :I + 1 :N
END

TO ABS :X
  IF :X < 0 OP - :X
  OP :X
END
```

?KLX 1 9 回车，即得图 1-7。

仿此，编制打印如图 1-8 所示的空心菱形。

?KLX 1 11

```
     **
    ****
   **  **
  **    **
 **      **
**        **
 **      **
  **    **
   **  **
    ****
     **
```

图 1-8

1-5 打印数字菱形

编程打印如图 1-9 所示的数字菱形。

```
         9
        999
       88888
      8888888
     777777777
    77777777777
   6666666666666
  666666666666666
 55555555555555555
5555555555555555555
 44444444444444444
  444444444444444
   3333333333333
    33333333333
     222222222
      2222222
       11111
        111
         0
```

图 1-9

解 分析此菱形的特征，每两行数字相同，末行除外，共有 19 行。为此，若令行数从 0 开始到 18，然后取 19 减去行数的差再被 2 除的整商，这样，一个奇数，一个偶数，它们的整商相同。就达到了相邻两行数字相同的要求。其余，只要按菱形解决每行的起始位置和每行个数就行了。

程序由三个过程组成。A 为主过程。有一个输入，从 0 开始。P、ABS 为子过程。ABS 输出一个数的绝对值，而 P 完成按定位打印。

程序如下：

程序 1-5

```
TO A :I; 0-18
  IF :I > 18 STOP
  IF :I > 9 MAKE "B 17-( :I-10)
* 2 P 1 A :I + 1 STOP
  MAKE "B :I * 2 + 1
  P 1
  A :I + 1
END

TO P :J
  IF :J > :B PR "'' STOP
  CURSOR ( ABS 9 - :I) + 10 + :J :I
  PRINT1 QUOTIENT 19 - :I 2
  P :J + 1
END
```

```
TO ABS :N
  IF :N < 0  OP - :N
  OP :N
END
```

? A 0 回车，即可得图 1-9。

1-6 打印夹行菱形

编程打印如图 1-10 所示的夹行菱形图案。

```
        @
      * @ *
    * * @ * *
  * * * @ * * *
@ @ @ @ @ @ @ @ @
  * * * @ * * *
    * * @ * *
      * @ *
        @
```

图 1-10

解 在 1-5 题菱形图案的基础上，只要作一些修改，即可完成此题。仔细观察，当 :I = (:N + 1)/2，即最中间的一行为@字符；又当 :J = (:N − 2 * (ABS(:N + 1)/2 − :I))/2 + 1，即每列最中间的一个字符为@。其余均为另一字符。故只要在打印前加以判断，区分是打印@还是 * 就行了。

程序如下：

程序 1-6

```
TO LXH :I :N ; 1-N
```

```
  IF :I > :N STOP
  CURSOR ( ABS ( :N + 1 ) / 2 - :I)
+ 1 :I
  PRT 1
  LXH :I + 1 :N
END

TO PRT :J
  IF :J > :N - 2 * ( ABS ( :N + 1 )
/ 2 - :I ) PR "" STOP
  IF ( ANYOF :J = ( INTEGER :N / 2 )
+ 1 - ( ABS ( :N + 1 ) / 2 - :I )
  :I = ( :N + 1 ) / 2 ) PRINT1 "@ PRT
:J + 1 STOP
  PRINT1 "*
  PRT :J + 1
END

TO ABS :X
  IF :X < 0 OP - :X
  OP :X
END
```

只要在命令状态键入LXH 1 21 回车，即可得如图1-10所示形式的夹行菱形。不过输入为1和21，得到的是一个共有21行的夹行菱形。

1-7 画正方形套

先画出一个正方形，连接这个正方形各边上的中点，又能画出一个正方形。再连接第二个正方形各边的中点，又能画出一个正方形，……如此类推，可以无限制地继续下去。要求设计 一 个 程序，能够根据输入数确定所画正 方形的个数，画出图形并计算图形中三角形的个数。譬 如，若以 6和第一个正方形边长为 100 作为输入，就能打印 出如图 1-11所示的正方形套并计算出三角形的个数为 20。

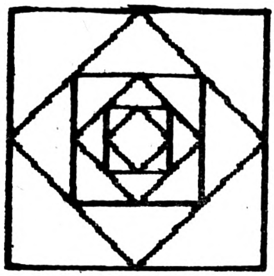


图 1-11

解 画图并不困难。问题在于统计三角形的个数。一个正方形，不能构成三角形。两个正方形，可有 4 个三角形，以后每增加一个正方形，就多 4 个三角形。所以三角形的个数与正方形个数有这样的 关系：$S=4(n-1)(n\geqslant 2)$，n 是正方形的个数。

用分屏作图模式，画出正方形套之后，在文本行显示出三角形的个数。

程序及运行示例如下：

程序 1-7

```
TO FSX :N :B
  DRAW
  REPEAT :N [REPEAT 4 [FD :B RT 90]
  FD :B/2 RT 45 MAKE "B (SQRT :B*:B
/2)] HT
  CURSOR 2 20 ( PR "S= ( :N - 1 ) *4 )
END

?FSX 6 100
  S= 20
```

即可得到如图 1-11 所示的正方形套，并 显示出 三角形的个数。

1-8　画“三菱”图形

编程绘制如图 1-12 所示的“三菱”徽标。要 求用 带输入的过程控制图形中每个菱形的边长和角度。

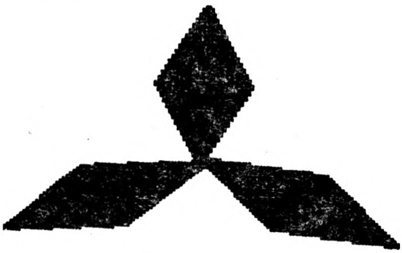


图　1-12

解 设计一个子过程 LXT 画出一个菱形，由输入L与 A 确定边长和夹角。然后边长逐次减 2 来填满中空。再设计主过程 SLT，输入同上。调用三次子过程，即可完成图形绘制。

程序及运行示例如下：

程序 1-8

```
TO SLT :L :A
  HT
  MAKE "H :L * COS :A / 2
  REPEAT 3 [LXT :L :A BK :H RT 120]
END

TO LXT :L :A
  IF :L = 0 STOP
  LT :A / 2
  REPEAT 2 [FD :L RT :A FD :L RT 18
0—:A]
  RT :A / 2 FD 2
  LXT :L - 2 :A
END
```

?SLT 20 30 回车，可得如图 1-12 的一个“三菱”形的图形。

此程序中，子过程也可以独立使用，画出一个菱形。

1-9 画 伞

编一个画伞程序。要求输入不同的伞把长、伞锥长及伞

张角，就可以画出如图 1-13 所示形状的伞来。

解 编一个带输入的过程，将伞把长、锥长及张角作为输入。使用三角函数命令，便可以计算出各部分的尺寸。程序定义两个过程，HS 为主过程，SJ为子过程。程序及运行示例如下：



图 1-13

程序 1-9

```
TO HS :C :B :A
  CS
  FD :C
  SJ :B * COS :A / 2 :A
  SJ :B * COS :A / 2 :A / 3 * 2
  SJ :B * COS :A / 2 :A / 3
  HT
END

TO SJ :H :A
  LT 180 - :A / 2
  FD :H / COS :A / 2
  LT 90 + :A / 2
  FD :H / ( COS :A / 2 ) * ( SIN :A
/ 2 ) * 2
  LT 90 + :A / 2
  FD :H /COS :A/2
  RT :A / 2
END
```

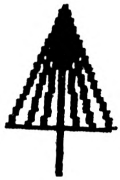


图 1-13.1

?HS 80 60 30 回车，可得图 1-13.1。

?HS 80 60 120 回车，可得图 1-13.2。



图 1-13.2

1-10 画 灯 笼

编一个画如图 1-14 所示的灯笼的程序。

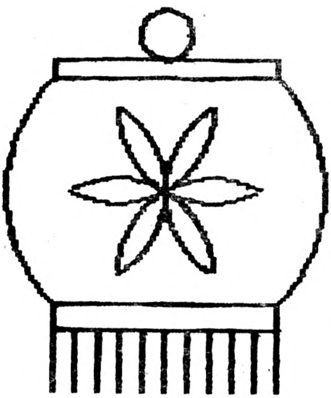


图 1-14

解 分析图形可以看出，灯笼是由小圆、两个矩形、两

段大圆弧、六段小圆弧（中心花瓣）及尾穗等几部分构成。

定义一个画弧子过程 HU，圆和弧便都可以画出。其余各部分及定位，都由主过程 DENG 直接完成。

程序如下：

程序 1-10

```
TO DENG
  FULLSCREEN PU SETY 50 PD
  .ASPECT .9
  LT 90
  REPEAT 36 [HU 5]
  REPEAT 2 [FD 25 LT 90 FD 5 LT 90 FD
25]
  BK 25 LT 90 FD 5 LT 60
  REPEAT 12 [HU 50]
  LT 60
  REPEAT 2 [FD 6 RT 90 FD 50 RT 90] FD
36
  REPEAT 10 [ BK 30 RT 90 FD 5 LT 90
FD 30 ]
  BK 36 RT 120
  REPEAT 12 [HU 50]
  PU RT 90 FD 25 PD
  LT 30
  REPEAT 3 [REPEAT 12 [HU 30] RT 120]
  LT 30 PU FD 30 PD RT 90
  REPEAT 3 [REPEAT 12 [HU 30] RT 120]
```

```
  HT
END

TO HU :R
  RT 5 FD 3.14 / 18 * :R RT 5
END
```

? DENG 回车，即得如图 1-14 所示灯笼。

1-11 画　　圆

编程绘画出如图 1-15 所示的图形。

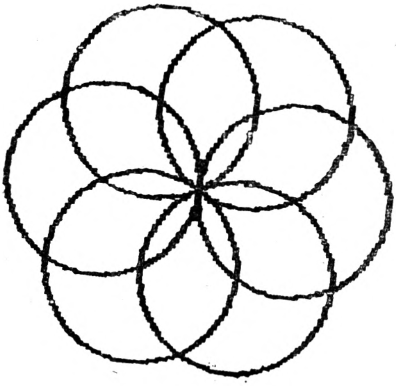


图 1-15

解法 1 分析图形可知，基本图形是圆，然后转过 360/6 度并重复画 6 次。

画圆最常见的是如下的过程：

```
  TO CIR :R
```

```
  REPEAT 36 [RT 5 FD :R*3.14/18 RT 5]
END
```

运行此过程，便可画出一个圆来。只要以半径为输入，无论海龟初始位置在何处，指向如何，都可画出如图 1-1 6 的圆。

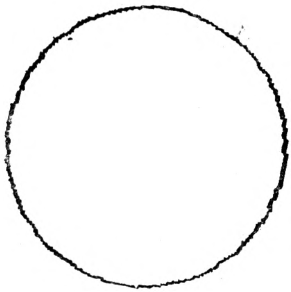


图 1-16

于是，建立主过程 HCIR，重复调用 CIR 六次即可。程序如下：

程序 1-11-1

```
TO HCIR :R
  REPEAT 6 [CIR :R RT 60]
  HT
END
```

键入 HCIR 50 回车，即得图 1-15 所示图形。

解法 2 建立一个由圆心位置、半径大小确定圆的过程，使用定位命令，相当于描点法画圆。过程为 YAN，输入 R

为半径，A、B 为圆心的 X 坐标及 Y 坐标。运行该过程，可以在坐标平面内任何地方画出指定大小的圆。

用主过程 HYAN 调用子过程 YAN，亦能画出图 1-15 的图形。

程序如下：

程序 1-11-2

```
TO YAN :R :A :B
  .ASPECT .9
  MAKE "T 0
  PU SETXY :A + :R * COS :T :B + :R *
SIN :T PD
  REPEAT 72 [SETXY :A + :R * COS :T :B +
:R * SIN :T MAKE "T :T + 5]
  HT
END

TO HYAN :R
  YAN :R :R 0 0
  YAN :R :R / 2 :R * .866 0
  YAN :R ( - :R/2 ) :R * .866 0
  YAN :R ( - :R ) 0 0
  YAN :R ( - :R / 2 ) ( - :R * .866 ) 0
  YAN :R :R /2 ( - :R * .866 ) 0
END
```

若对画圆子过程 YAN 稍加修改，还可以用来画弧段。具体方法是删去过程中的第二行命令。增加两个输入，用 T

表示开始角度，Z 表示结束的角度。并将重复次数 72 改为 (:Z - :T)/5，便可以在平面内画已知中心及半径的圆弧。

1-12 画 椭 圆

若已知椭圆的中心坐标位置，长半轴和短半轴及长轴的倾斜角(长轴与平面直角坐标系 x 轴的夹角)。编一个画椭圆的程序。

解 ① 长短半轴分别平行于 x 轴与 y 轴的椭圆程序。

根据椭圆参数方程

$$\begin{cases} x = a\cos\theta \\ y = b\sin\theta \end{cases}$$

可以求出椭圆上任何一点的坐标 x、y。重复使用此公式，每隔 5° 求一个点，然后依次连结，相当于描点法画图。

为此，建立过程 TY1，带六个输入。A、B 为椭圆中心坐标，C、D 为长、短半轴的长，Q 为起始角度，Z 为终止角度。

程序及运行示例如下：

程序 1-12-1

```
TO TY1 :A :B :C :D :Q :Z
  PU
  SETXY :A + :C * COS :Q :B + :D * SI
N :Q PD
  REPEAT ( :Z - :Q ) / 5 [SETXY :A+:C
*COS :Q :B+:D*SIN :Q MAKE "Q :Q+5]
  HT PU HOME PD
```

END

若 键入 TY1 40 40 50 30 0 360 回 车，则可画出中心在(40，40)，长半轴为50，短半轴为 30 的椭圆(如图 1-17)。

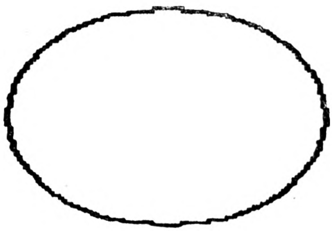


图 1-17

② 长轴倾斜的椭圆。即长轴与 x 坐标轴有一定 夹 角。设椭圆倾斜角为 T，并以 T 也为一个输 入。应用 坐 标平面内的旋转轴公式，可以分别求出椭圆 上 某一点的 x、y 坐标，然后用定位命令，逐个算出各个点并画图。

程序如下：

程序 1-12-2

```
TO TY2 :A :B :C :D :Q :Z :T
  PU
  SETXY :A + :C * (COS :Q ) * ( COS
:T ) - :D * ( SIN :Q ) * SIN :T :B + :C
* ( COS :Q ) * ( SIN :T ) + :D * (SIN :Q )
* COS :T PD
  REPEAT ( :Z - ;Q ) / 5 [SETXY :A+:C
*(COS :Q) * (COS :T)-:D*(SIN :Q)*
```

```
SIN :T :B+:C* (COS :Q) * (SIN :T)+:D
* (SIN :Q) * COS :T MAKE "Q :Q+5 ]
  PU HOME PD HT
END
```

以上两个程序中的 Q 与 Z，从 0 到 360，可画出椭圆。若从任一角度开始，到另一角度结束，则可画出椭圆的一部分弧段。但要注意使 Q<Z，且 Q、Z 均可正可负。

当输入中 C 与 D 相同时，则可画出圆。

1-13 画 三 叉 树

编制一个画形如图 1-18 所示的三叉树的程序。要求用递归调用过程形式。

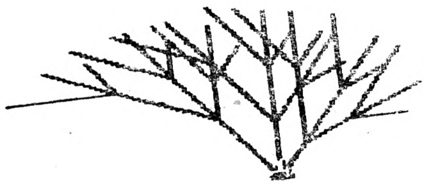


图 1-18

解 以 V 形为基本树型的二叉树程序，是典型的递归范例，几乎所有介绍 LOGO 的书上都可见到。现在所要画的三叉树，基本树型变成了三枝，也就是说，整株树都是由三枝树型及其每个顶点处又由较小的三枝树型所组成。于是，基于同样的道理，我们在编制完成一个基本树型的递归过程时，只要多递归调用一次就行了。

程序如下。过程名为 TREE，带三个输入。L 表示首次基树枝的长，A 表示三枝之间两两的夹角，D 表示整树的层数，即“深度”。若要画左枝长为 20、两枝夹角为 30° 的 3 层三叉树，只要执行

?TREE 20 30 3 回车，就会得图 1-18 所示的结果。

程序 1-13

```
TO TREE :L :A :D
  IF :D = 0 STOP
  LT :A
  FD :L * 2
  TREE :L :A :D - 1
  BK :L * 2
  RT :A
  FD :L * 1.5
  TREE :L :A :D - 1
  BK :L * 1.5
  RT :A
  FD :L
  TREE :L :A :D -1
  BK :L
  LT :A
END
```

运行程序之前，对海龟起始位置及指向不加限制。执行之后，画笔又回到开始的位置及原来的方向。

二 常用过程

2-1 字函数过程（7条）

对应于 BAS1C 语言中的字符串函数，建立如下几个 LOGO 字函数过程。

① 求字长。

② 输出字中指定字符。

③ 输出从右数起的指定字符。

④ 取左子字函数。

⑤ 取右子字函数。

⑥ 取中间子字函数。

⑦ 取从右数起的中间子字函数。

解 BAS1C 语言中的字符串概念及其函数，使它广泛地应用于事务管理及非数值性计算工作。LOGO 中无此类原语命令，仿照建立几个 LOGO 字函数过程，随时调用，则更增添 LOGO 字处理功能的灵活性。

1．求字长，即 LOGO 字中字符的个数。见 LEN 过程。如键入 PR LEN ″ABCDE 回车，显示 5。

2．输出字中指定字符。如键入 PR PICK ″ABCDE 3 回车，显示 C。见过程 PICK。

3．输出字中从右边数起的第几个字符。见过程 PICKR。如键入 PR PICKR ″ABCDE 2 回车，显示D。

4．取左子字函数，见LEFT$过程。带两个输 入。等同 于 BASIC 中 的 LEFT$(A$，n) 函 数。如 键 入 PR LEFT$ "ABCDE 3 回车，显示 ABC。

5．取右子字函数，见 RIGHT$过程。带 两个 输入。等同于 BASIC 中的 RIGHT$(A$,n)函数。如键 入 PR RIGHT$ "ABCDE 3 回车，显示 CDE。

6．取中间子字函数，见 MID$过程。带三个输入。一个 LOGO 字，两个数。概念完全等同于 BASIC 中 求 子字符串的 MID$(A$，n，m) 函 数。例 如 键 入 PR MID$ "ABCDE 2 3 回车，显示 BCD。当 I=1 时，等效于 4。

7．取从右数起第几个开始的几个字符子字。类似 于 6。当 LOGO 字很长，欲取子字又很偏后时，此 过 程比 6 来得快。当 I=1 时，等效于 5。例如键入 PR MIDR$ "ABCDE 3 2 回车，显示 BC。见 MIDR$过程。

七个过程清单如下：

程序 2-1

```
TO LEN :W
  IF :W = "''OP 0
  OP 1+LEN BF :W
END

TO PICK :A$ :N
  IF :N=1 OP FIRST :A$
  OP PICK BF :A$ :N-1
END

TO PICKR :A$ :I
```

```
  IF :I = 1 OP LAST :A$
  OP PICKR BL :A$ :I-1
END

TO MID$ :A$ :I :J
  IF :J = 1 OP PICK :A$ :I
  OP WORD PICK :A$ :I MID$ :A$ :I
 +1 :J-1
END

TO LEFT$ :A$ :I
  IF :I = 1 OP FIRST :A$
  OP WORD FIRST :A$ LEFT$ BF :A$ :I
-1
END

TO MIDR$ :A$ :I :J
  IF :J = 1 OP PICKR :A$ :I
  OP WORD MIDR$ :A$ :I+1 :J-1 P
ICKR :A$ :I
END

TO RIGHT$ :A$ :I
  IF :I=1 OP LAST :A$
  OP WORD RIGHT$ BL :A$ :I-1 LAST
:A$
END
```

2-2 表函数过程（8条）

仿照 LOGO 字函数过程，建立处理表的几个常用函数过程。

① 求表长。

② 是否为表的成员。

③ 输出表中指定项。

④ 输出从右数起的表中指定项。

⑤ 取左子表过程。

⑥ 取右子表过程。

⑦ 取中间子表过程。

⑧ 取从右数起的中间子表过程。

解 这几个表处理的常用过程，类似于字函数过程。随时根据需要调用，使编程灵活熟练。下面是 8 个过程清单：

程序 2-2

```
TO LNL :L
  IF :L=[] OP 0
  OP 1+LNL BF :L
END

TO MEL :W :L
  IF :L=[] OP "F
  IF :W=FIRST :L OP "T
  OP MEL :W BF :L
END
```

```
TO PLL :L :I
  IF :I = 1 OP FIRST :L
  OP PLL BF :L :I-1
END

TO PRL :L :I
  IF :I = 1 OP LAST :L
  OP PRL BL :L :I-1
END

TO LTL :L :I
  IF :I = 1 OP FIRST :L
  OP SE FIRST :L LTL BF :L :I - 1
END

TO RTL :L :I
  IF :I = 1 OP LAST :L
  OP SE RTL BL :L :I - 1 LAST :L
END

TO MLL :L :I :J
  IF :J = 1 OP PLL :L :I
  OP SE PLL :L :I MLL :L :I + 1 :J
  -1
END

TO MRL :L :I :J
  IF :J = 1 OP PRL :L :I
  OP SE MRL :L :I +1 :J -1 PRL :L :I
```

END

以上各过程的意义及其使用方法如下：

1. LNL 过程，计表长。以表作为输入，输出的是项数。即表中所含元素的个数。如，键入 PR LNL [A B C D E]回车，结果显示 5。

2. MEL 过程，判断字或数或表是否为表的元素。以一个字（或数或表）及表为输入。输出为真（TRUE）或假（FALSE）。如，键入 PR MEL "A [A B C D E]回车，结果显示T；键入 PR MEL 5 [1 2 3 4]回车，结果显示 F。

3. PLL 过程，输出表中指定元素。以一个表及数为输入，输出表中第几个元素。如，键入 PR PLL [A B C D E] 2回车，结果显示 B。

4. PRL 过程，类似 3，不过表中元素是从右向左数。如，键入 PR [A B C D E] 2 回车，结果显示 D。

5. LTL 过程，从左取子表过程。以一个表及数为输入，输出表中从左开始的几项元素。如，键入 PR LTL [A B C D E] 3 回车，结果显示 A B C。

6. RTL 过程，类似 5，从右取子表过程，输出表中从右数起的几项元素。如，键入 PR RTL [A B C D E] 3 回车，结果显示 C D E。

7. MLL 过程，取中间子表过程。以一个表及两个数为其输入，输出表中从第几个元素开始的几个元素。如，键入 PR MLL [A B C D E] 3 2 回车，结果显示C D。

8. MRL 过程，类似 7，但表从右数起。如，键入PR MRL [A B C D E] 3 2 回车，结果显示 B C。

2-3 计算常用的函数过程（15条）

在 LOGO 数值计算中经常使用的一些函数过程：

① 绝对值函数过程。

② 符号函数过程。

③ 乘方运算函数过程。

④ 阶乘函数过程。

⑤ 指数函数过程。

⑥ 正切函数过程。

⑦ 自然数累加函数过程。

⑧ 求和函数过程（任意多个加数）。

⑨ 计数函数过程。

⑩ 求均值函数过程。

⑪ 求二数最大公约数函数过程。

⑫ 求二数最小公倍数函数过程。

⑬ 求整函数过程。

⑭ 求纯小数函数过程。

⑮ 舍入操作函数过程。

下面是以上 15 条共 16 个过程清单。LOGO 在数值计算问题中，是可以根据实际情况，随时调用以上各个函数过程的。当然，它们也可以独立使用。

程序 2-3

```
TO ABS :X
  IF :X<0 OP -:X
  OP :X
```

```
END

TO SGN :X
  IF :X=0 OP 0
  IF :X>0 OP 1
  OP -1
END

TO POW :A :X
  IF :X=0 OP 1
  IF :X=1 OP :A
  IF :X=-1 OP 1/ :A
  IF :X>1 OP :A * POW :A :X-1
  IF :X<-1 OP (1/ :A)*(POW :A :X+1)
  OP [NOT INTEGRE]
END

TO FAC :N
  IF ANYOF :N = 0 :N = 1 OP 1
  OP :N * FAC :N-1
END

TO EXP :X
  MAKE "T 1
  MAKE "N 1
  IF :X = 0 OP 1
  IF :X > 0 OP EXP.1 :N :X
```

```
  MAKE "X ( - :X)
  OP 1/EXP.1 :N :X
END

TO EXP.1 :I :X
  MAKE "T :T * :X/ :I
  IF :T < 9.99991N7 OP 1
  OP :T+EXP.1 :I+1 :X
END

TO TAN :X
  IF :X = 90 OP [NOT DEFIND]
  OP (SIN :X) / COS :X
END

TO SUM :N
  MAKE "S 0
  AA: IF :N=0 OP :S
  MAKE "S :S + :N
  MAKE "N :N -1
  GO "AA
END

TO ADD :L
  MAKE "S 0
  AA: IF :L = [ ] OP :S
  MAKE "S :S+FIRST :L
  MAKE "L BF :L
```

```
  GO "AA
END

TO CON :L
  MAKE "G 0
  AA: IF :L=[  ] OP :G
  MAKE "G :G+1
  MAKE "L BF :L
  GO "AA
END

TO AVE :L
  IF :L=[  ] OP 0
  OP (ADD :L) / CON :L
END

TO GCD :M :N
  IF REMAINDER :M :N=0 OP :N
  OP GCD :N REMAINDER :M :N
END

TO LCM :M :N
  OP :M * :N/GCD :M :N
END

TO INT :X
  IF :X < 0 OP ROUND ( :X - .5)
  OP INTEGER :X
END
```

```
TO FRA :X
  OP :X - INTEGER :X
END

TO ROF :N :P
  OP (ROUND (:N * POW 10 :P)) /
  POW 10 :P
END
```

对于它们的意义及用法，下面分别加以说明。

1. ABS.绝对值函数过程。求|x|。如键入PR ABS 5回车，结果为5。如键入 PR ABS -5回车，结果仍为5。

2. SGN. 符号函数过程。即数学上的

$$\text{signx}=\begin{cases}1 & \text{当 } x>0,\\ 0 & \text{当 } x=0,\\ -1 & \text{当 } x<0.\end{cases}$$

函数。如键入PR SGN 5回车，结果为1；如键入PR SGN -5回车，结果为-1；如键入PR SGN 0回车，结果为0。

3. POW. 乘方运算函数过程。求 a^x。有两个输入。前一个为底数，后一个为指数。底数可以是任意实数，指数可正可负，但必须是整数。对于0的正整数次幂，结果为0。对于0的负整数次幂，不加定义，机器会自动给出错误信息。因为过程中有分式运算，0不能作除数。

如键入PR POW 2 3回车，显示8；如键入PR POW 2 (-3)回车，显示.125。

4. FAC. 求阶乘函数过程，即求n!。输入一个N,

即可求它的阶乘。应当说明的是，在CEC-I LOGO及APPLE-Ⅱ机的MIT LOGO中，最大只能求33! 即n≤33。

如键入PR FAC 10回车，显示3628800。

5. EXP.指数函数过程，求e^x。在此，e=2.718281。x可正可负。但必须$|x|\leq 19$。它还以EXP.1为其子过程。由于这个函数过程应用数学上的近似计算公式

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots + \frac{x^n}{n!} + \cdots$$

所以精确度取小于10^{-6}(1.N6)的项略去不计，即过程中的9.99991N7。

如键入PR EXP 0回车，显示1。如键入PR EXP 1回车，显示2.71828。如键入PR EXP 19回车，显示5.60282N9。

6. TAN. 正切函数过程，求某角的正切函数，即tgx。LOGO中三角函数的原语命令只有三个，正弦、余弦和反正切。仿此，还可以定义其它更多的三角函数过程。在使用此函数过程时，其输入必须是角度值。

如键入PR TAN 30回车，显示0.5773…。如键入PR TAN 45回车，显示1.。

7. SUM. 求自然数累加和函数过程。

如键入PR SUM 100回车，显示5050。若键入PR SUM 10000回车，结果为50005000。

8. ADD. 求加数和函数过程。以表为输入，表中各项元素是欲求和的各个加数。

如键入PR ADD [3 5 7 9 10]回车，显示34。

9. CON．计数函数过程。以表为输入，统计表中元素的个数。类似于表处理函数过程及字函数过程中的 LEN。

如键入 PR CON [1 1 1 1 1] 回车，显示 5。

10. AVE．求平均值函数过程。以表为输入，表中各项元素必须是数。它以 ADD 及 CON 为子过程。直接调用它们而输出其商。

如键入 PR AVE [3.5 4.7 1.4 2.8]回车，显示3.1。

11. GCD．求二数最大公约数函数过程。在此用辗转相除法。该过程带两个输入，大小与次序无关。

如键入 PR GCD 25 75 回车，显示 25。

12. LCM．求二数最小公倍数函数过程。二数相乘被它们的最大公约数除，则得最小公倍数。所以此过程又以 GCD 为子过程。

如键入 PR LCM 7 9 回车，显示 63。

13. INT．整数函数过程。同数学上的[x]，即求不大于 x 的最大整数。与 BASIC 中的 INT(X)函数等同。但不同于 LOGO 中的原语函数命令 INTEGER 及 ROUND。这两个命令前者是截尾取整，后者是舍入取整(四舍五入)。

如键入 PR INT 3.47 回车，显示 3。如键入PR INT -4.62 回车，显示-5。

14. FRA．取纯小数函数过程，即求{x}。无论正、负，都能输出所给数的纯小数部分。应说明的是，由于机器对小数部分的不同进制的转换，可能产生一定的误差。

如键入 PR FRA 1.234 回车，显示.234。如键入 PR FRA 1.00012 回车，显示 1.19686N4。有一定的误差。就是说，不是 0.00012，而是 0.000119686。

15. ROF. 舍入操作函数过程。按指定位数留取小数部分的函数过程。可以进行四舍五入操作。该过程带两个输入。第一个是要操作的数。第二个是保留的位数。可正也可负。正数表示从小数点往后数的位数。负数表示从小数点往前数的位数。但必须是整数。该过程调用 POW 为子过程。

如键入PR ROF 1.55555 2回车，显示1.56。如键入PR ROF 123.123 0回车，显示123.。如键入PR ROF 1345.67 (-2)回车，显示1300.。

2-4 FOR循环过程

建立LOGO FOR循环常用过程。

解 用RUN命令运行由表操作构造的表，这是为基本LOGO增加了一种新的程序设计结构。下面我们建立FOR循环过程，对具有循环性质的这一类型问题，在知道循环控制变量、初值、终值、步长及需要不断重复执行的循环体的情况下，都可以调用该过程。从而使编程变得简便易行。

解法1 建立主过程FOR1，它有两个表作输入，第二个表是循环体表，第一个表是由循环控制变量、初值、终值和步长组成的表。再建立子过程V、I、F、S分别输出表的指定项。及子过程FORL1，处理正步长，子过程FORL 2，处理负步长。程序清单如下。

程序 2-4-1

```
TO FOR1 :L1 :L2
  IF ( I :L1 ) >( F :L1 ) FORL2(V :L1)
(I :L1 )( F :L1 )( S :L1 ) :L2 STOP
```

```
  FORL1 (V :L1) (I :L1)(F :L1)(S :L1) :L2
END

TO V :L
  OP FIRST :L
END

TO I :L
  OP FIRST BF :L
END

TO F :L
  OP LAST BL :L
END

TO S :L
  OP LAST :L
END

TO FORL1 :V :I :F :S :L2
  IF :I> :F STOP
  MAKE :V :I
  RUN :L2
  FORL1 :V (:I+ :S) :F :S :L2
END

TO FORL2 :V :I :F :S :L2
  IF :I< :F STOP
  MAKE :V :I
  RUN :L2
```

```
  FORL2 :V (:I+ :S) :F :S :L2
END
```

解法 2 仿解法1，删去 V、I、F、S 四个过程，而将表中指定项，直接用 MAKE 命令赋给以上四个同名变量。其它保持不变。这样只要三个过程。主过程FOR2，子过程 FORL1及 FORL2 同上。程序清单如下。

程序2-4-2

```
TO FOR2 :L1 :L2
  MAKE "V (FIRST :L1)
  MAKE "I (FIRST BF :L1)
  MAKE "F (LAST BL :L1)
  MAKE "S (LAST :L1)
  IF :I> :F FORL2 :V :I :F :S :L2
STOP
  FORL1 :V :I :F :S :L2
END
```

由于 FOR2 中使用了 MAKE 命令，此法却不能用来解决二重及多重循环，即循环嵌套问题。

解法 3 为了能解决循环嵌套，即二重、三重甚至多重循环问题，又不使用很多个过程，用 GO 命令设置过程本身中的循环模块，则只用一个 FOR过程，即可完成定义循环过程。程序如下。不过要将第一个输入表，换作单个的控制变量、初值、终值、步长为输入，循环体表照旧。

程序2-4-3

```
TO FOR :V :I :F :S :L2
  IF :I > :F GO "BB
```

```
  AA: IF :I > :F STOP
  MAKE :V :I
  RUN :L2
  MAKE "I :I + :S
  GO "AA
  BB: IF :I < :F STOP
  MAKE :V :I
  RUN :L2
  MAKE "I :I + :S
  GO "BB
END
```

对于我们所定义的FOR循环常用过程，无论步长是正是负，是 1 非 1，及初值终值，谁大谁小，都可解决。而且，无论是单循环，二重循环，多重循环，都可嵌套调用。

今后可以看到，FOR 循环常用过程的应用将是十分广泛的。

2-5 一维数组

在LOGO中建立一维数组的程序。

解 LOGO语言中，没有数组的概念和下标变量的设置。但是其赋值命令 MAKE 功能却很强。我们可以通过复合字命令WORD及MAKE，建立LOGO中的一维数组形式。

过程 SZI1，带一个输入N，表示数组元素的个数。当键入过程名及输入并回车后，屏幕提示"请输入元素"，这时候计算机将等待你从键盘上一个个地输入数据。最后打印出

含有N个元素的数组及其值。

程序及运行示例如下。

程序2-5-1

```
TO SZI1 :N ; A(N)
  MAKE "I 0 PR [QING SHU RU YUAN SO
U!]
  REPEAT :N [MAKE WORD "A :I FIRST
RQ MAKE "I :I+1]
  MAKE "I 0
  REPEAT :N [(PRINT1 WORD "A :I "=
THING WORD "A :I) PR "" MAKE "I :
+1]
END

?SZI1 5
QING SHU RU YUAN SOU!
A0=101
A1=102
A2=103
A3=104
A4=105
```

程序SZR1是带两个输入的过程。第一个是数组元素的个数，第二个则是让机器随机产生元素值的范围。键入SZR1及其两个输入并回车后，即可得到一个一维数组N个元素及其在R之内的值。

程序及运行示例如下。

程序2-5-2

```
TO SZR1 :N :R ; A(N) <R
  MAKE "I 0
  REPEAT :N [MAKE WORD "A :I
RANDOM :R MAKE "I :I+1]
  MAKE "I 0
  REPEAT :N [(PRINT1 WORD "A :I "=
THING WORD "A :I ) PR "" MAKE "I
:I+1]
END
```

```
?SZR1 10 101
A0=55
A1=46
A2=64
A3=72
A4=30
A5=21
A6=37
A7=6
A8=79
A9=54
```

用如此数组形式编程，在许多场合是很有用的。如统计学生成绩分段情况，这样便可以使程序十分简练。

2-6 二 维 数 组

在LOGO中建立二维数组的程序。

解 如同建立一维数组一样，可以在LOGO中建立二维数组。不过由于需要两个下标，形式要复杂一些罢了。

过程 SHZ2，带三个输入。前两个 N、M表示二维的两个最大下标。第三个为随机数值范围。运行该过程将打印出某个范围内值的二维数组。

程序及运行示例如下：

程序2-6

```
TO SHZ2 :N :M :R ; A(N,M)
  MAKE "I 0
  REPEAT :N [MAKE "J 0 REPEAT :M
[MAKE (WORD "A( :I ", :J ")) RANDOM
:R PRINT1 (WORD "A( :I ", :J ") "= )
PR THING (WORD "A( :I ", :J ") )
MAKE "J :J+1] MAKE "I :I+1]
END

?SHZ2 3 4 50
A(0,0) = 31
A(0,1) = 0
A(0,2) = 3
A(0,3) = 49
A(1,0) = 23
```

A(1,1) = 5

A(1,2) = 11

A(1,3) = 6

A(2,0) = 47

A(2,1) = 46

A(2,2) = 15

A(2,3) = 26

在具体的实际应用问题中，你所建立的过程形式，也许与此不同。但仿此原理，是可以建立自己所需的二维数组的过程形式。

三 数值计算

3-1 求自然数的和

编程求n个自然数的累加和。

解 问题很简单。我们可以用如下几种不同的方法解决。

法1 直接使用重复命令。重复次数由加数的个数N决定，把它作为输入。另设变量J作加数(或记数器)，变量S存放累加和。

程序及其运行示例如下：

程序3-1-1

```
TO LJ1 :N
  MAKE "S 0
  MAKE "J 0
  REPEAT :N [MAKE "J :J+1 MAKE "S
:S+:J]
  PR :S
END

?LJ1 100
5050
```

法2 在过程中使用转向命令构成循环。过程输入与变量同法1。

程序及运行示例如下：

程序3-1-2

```
TO LJ2 :N
  MAKE "S 0
  MAKE "J 1
  AA: MAKE "S :S + :J
  IF :J = :N PR :S STOP
  MAKE "J :J + 1
  GO "AA
END

?LJ2 100
5050
```

法 3　利用等差数列的求和公式，在过程中直接使用表达式求和。

程序及运行示例如下：

程序3-1-3

```
TO LJ3 :N
  MAKE "S ( 1 + :N ) / 2 * :N
  ( PR "S= :S )
END

?LJ3 100
S = 5050.

?LJ3 1000
S =  500500.
```

法 4　其次，我们还可以使用FOR循环常用过程，由主

过程 LJ4 调用。同样可以完成累加求和。控制变量为 K,初值1，终值为 N，步长为1。

程序及运行结果如下：

程序3-1-4

```
TO LJ4 :N
  MAKE "T 0
  FOR "K 1 :N 1 [MAKE "T :T+:K]
  ( PR "T= :T )
END

?LJ4 100
T= 5050
```

可以看出,对于重复和循环性的工作,编程首先想到的就是使用重复命令。转向命令可以克服直接递归深度的限制。采用公式法,是最简单的,但若无法推导出计算公式，就无能为力了。能使用 FOR 过程的,一般用 REPEAT 也可以完成。

尚需指出的是，求 n个自然数的和

$$S=n+(n-1)+(n-2)+\cdots+3+2+1$$

若用 S(I)记前I个自然数之和，一般有

$$S(I)=I+S(I-1)$$

正好贯穿着一个递归思想。因而，对于初学者可能编写出如下的递归函数过程：

```
TO LJ5 :N
  IF :N = 0 OP 0
  OP :N + LJ5 :N - 1
END
```

或者引入参变量作为输入，I记个数，S记和，N同前，编出如下递归过程：

```
TO LJ6 :I :N :S
  IF :I > :N OP :S
  OP LJ6 :I + 1 :N :S + :I
END
```

这两个过程，虽然十分简单明了，但是，由于递归深度的限制及递归运算速度较慢，一般并不采用。如前者只能演算 67 层，后者也只能演算 73 层。这是应该注意的。

3-2 求若干个加数的和

求已知的若干个加数之和。要求打印出算式及其结果。

解法 1 编制两个过程，QHE 及 JHE，键入主过程名 QHE并回车后，再由键盘输入各个加数存表变量 L中。然后由子过程 JHE 计算并打印算式及结果。其中M记各个加数，J记总和。程序还可以连续计算，如欲结束，则只要输入 E为终止标志，返回命令层。

程序及运行示例如下：

程序 3-2-1

```
TO QHE
  MAKE "L RQ
  IF FIRST :L = "E STOP
  MAKE "J 0
  JHE :L
  REPEAT 1300 [ ] QHE
```

```
END

TO JHE :N
  MAKE "M FIRST :N
  MAKE "J :J + :M
  PRINT1 :M
  IF NOT BF :N = [ ] PRINT1 "+ ELSE
PRINT1 "= PR :J STOP
  JHE BF :N
END

?QHE
1+2+3+4+5+6+7+8+9=45
98765+43210=141975
```

解法 2 将各加数以表的形式，直接作为主过程 QHE 1 的输入，则更为简练。子过程JHE1，变量J记和。变量M从加数表中逐个取出加数进行运算。

程序及运行示例如下：

程序 3-2-2

```
TO QHE1 :L
  IF :L = [ ] STOP
  MAKE "J 0
  JHE1 :L
END

TO JHE1 :N
  MAKE "M FIRST :N
  MAKE "J :J + :M
```

```
  PRINT1 :M
  IF NOT BF :N = [ ] PRINT1 "+ ELSE
PRINT1 "= PR :J STOP
  JHE1 BF :N
END

?QHE1 [ 1 2 3 4 5 6 7 8 9 ]
1+2+3+4+5+6+7+8+9=45

?QHE1 [9876 54321]
9876+54321=64197
```

3-3　求 T＝N!

编程求阶乘T＝N!。

解　$T=N!=N(N-1)(N-2)\cdots3\times2\times1$

可知

$$
\begin{aligned}
T=\ & T_n=n\times T_{n-1}\\
& T_{n-1}=(n-1)\times T_{n-2}\\
& \quad\vdots\\
& T_2=2\times T_1\\
& T_1=1\times1
\end{aligned}
$$

故，可用如下不同的方法编制程序。

法 1　直接建立递归过程。其输入即要求的N。

程序及运行示例如下（需注意，它是个函数过程，运行时前面应加命令PR）：

程序 3-3-1

```
TO JC1 :N
  IF :N = 1 OP 1
  OP :N * JC1 :N - 1
END

?PR JC1 10
3628800
```

法 2 使用重复命令。N为输入，变量 I 为计数器，也即是相乘的因数，T 存放累乘结果。

程序及运行示例如下：

程序3-3-2

```
TO JC2 :N
  MAKE "I 0
  MAKE "T 1
  REPEAT :N [MAKE "I :I+1 MAKE "T
:T*:I]
  PR :T
END

?JC2 5
120
```

法 3 使用转向命令设置循环。

程序及运行示例如下：

程序 3-3-3

```
TO JC3 :N
  MAKE "I 1
```

```
  MAKE "T 1
  AA: IF :I > :N PR :T STOP
  MAKE "T :T * :I
  MAKE "I :I + 1
  GO "AA
END

?JC3 8
40320
```

法 4　可以调用FOR过程。

程序及运行示例如下：

程序 3-3-4

```
TO JC4 :N
  MAKE "T 1
  FOR "K 1 :N 1 [MAKE "T :T*:K]
  PR :T
END

?JC4 5
120
```

3-4　父子年龄问题

一个年龄在 13 至 19 岁的孩子，若把他自己的年龄写在他父亲年龄的后面，则得到一个四位数。从这个数中减去他们年龄的差得到4289。求他们父子各自的年龄。

解　利用复合字把他们的年龄并在一起，并赋给变量P，

然后判断。建立两个过程。为了减少递归调用次数，子过程ZZ以孩子年龄为输入，在13～19之间，逐次增1。主过程FF以父亲年龄为输入，最小不会小于33岁。一旦求出符合条件的两个年龄，即用TOPLEVEL返回命令层。

程序及运行结果如下：

程序 3-4

```
TO FF :M ; 33-
  ZZ 13
  FF :M + 1
END

TO ZZ :N ; 13-19
  IF :N > 19 STOP
  MAKE "P WORD :M :N
  IF :P - ( :M - :N ) = 4289 ( PR
"F= :M ) ( PR "Z= :N ) TOPLEVEL
  ZZ :N + 1
END

?FF 33
F= 43
Z= 16
```

3-5　求S＝1！＋2！＋3！＋…＋10！的值

编程求 S＝1！＋2！＋3！＋…＋10！的值。

解　可以看出，项数 x、每项的值 T 及前 x项的和有如

下关系：

项数 x	各项的值T	前 x 项之和
1	$T_1=1$	$S_1=1$
2	$T_2=T_1\times 2$	$S_2=S_1+T_2$
3	$T_3=T_2\times 3$	$S_3=S_2+T_3$
⋮	⋮	⋮
10	$T_{10}=T_9\times 10$	$S_{10}=S_9+T_{10}$

只要设置变量 T 记各项的值，变量 S 记总和，过程输入 X 控制项数。

法 1 主过程 JIA 给变量 T 及 S 赋初值 1 和0。子过程 CHE 由 1 开始，递归 10 次使变量 T 连乘，同时使 S 与 T 连加。

程序及运行结果如下：

程序 3-5-1

```
TO JIA
  MAKE "S 0
  MAKE "T 1
  CHE 1
END

TO CHE :X
  IF :X > 10 ( PR "S= :S ) STOP
  MAKE "T :T * :X
  MAKE "S :S + :T
  CHE :X + 1
END
```

```
?JIA
S= 4037913
```

法 2　用重复命令。输入N控制重复次数。变量Ⅰ记项，变量J记各项的阶乘值，变量S记总和。只用一个过程JCX。

程序 3-5-2

```
TO JCX :N
  MAKE "I 0
  MAKE "J 1
  MAKE "S 0
  REPEAT :N [MAKE "I :I+1 MAKE "J
:J*:I MAKE "S :S +:J]
  ( PR "S= :S )
END

?JCX 10
S= 4037913
```

3-6　求分式值①

编程求 $S=1+\frac{1}{1\times 2}+\frac{1}{2\times 3}+\frac{1}{3\times 4}+\cdots$

$+\frac{1}{n(n+1)}$的值。

解　可以看出，除第一项之外，其余各项分母均可看成项数乘以(项数+1)。项数从1到N。

法 1　建立过程QZT1，带三个输入，Ⅰ记项数，从1开始；N为总项数；S记总和，从1开始。当Ⅰ大于N时，打印

输出 S的值。

程序及运行结果如下（求出前 101 项之值）：

程序 3-6-1

```
TO QZT1 :I :N :S ; 1 N 1
  IF :I > :N ( PR "S= :S ) STOP
  QZT1 :I + 1 :N :S + 1 / ( :I * (:I
+ 1 ) )
END

?QZT1 1 100 1
S= 1.99009
```

法 2 将指定项数N进行计算，改为按精度要求计算，即当计算到某一项时，该项之值小于指定的数E，如E = 10^{-6}，则其后各项统统略去不计。此时，只需建立过程 QZT3，也有三个输入。I 记项数，从 1 开始，E 为精度，取 1.N 6，S记总和，从 1 开始。

程序及运行结果如下：

程序 3-6-2

```
TO QZT3 :I :E :S ;  1 1.N 6 1
  MAKE "T :I * ( :I + 1 )
  MAKE "T 1 / :T
  IF :T < :E ( PR "S= :S ) STOP
  QZT3 :I + 1 :E :S + :T
END

?QZT3 1 9.99999N7 1
S= 1.99888
```

3-7 求分式值②

编程求算式 $S=1+\frac{1}{1+\frac{1}{2}}+\frac{1}{1+\frac{1}{2}+\frac{1}{3}}+\cdots+\frac{1}{1+\frac{1}{2}+\cdots+\frac{1}{n}}$的值。

解 粗看算式繁难，无从下手。其实很有规律，类似于上题。只要设置三个变量，S、K、I，分别用来记项数，记每项分母上分式之值，及整个分式之和就行了。

法 1 定义过程FSH，带一个输入，指明要计算的项数。程序及运行结果如下。可以比较 5 项及 10 项的情况。

程序 3-7-1

```
TO FSH :N
  MAKE "S 0 MAKE "K 0 MAKE "I 0
  REPEAT :N [MAKE "I :I+1 MAKE "K
:K+1/:I MAKE "S :S+1/:K] (PR "S= :S)
END

? FSH 10
S= 4.98675

? FSH 100
S= 25.5298
```

法 2 定义过程FSH1，带一个输入N。设置变量I记

项数，K 记各项之值，S 记总和。用转向命令重复 N 次。

程序及运行示例如下：

程序 3-7-2

```
TO FSH1 :N
  MAKE "K 0 MAKE "S 0 MAKE "I 1
  AA: IF :I > :N (PR "S= :S) STOP
  MAKE "K :K + 1 / :I
  MAKE "S :S + 1 / :K
  MAKE "I :I + 1 GO "AA
END

? FSH1 10
S= 4.98675
```

法 3 定义 FSH2 过程调用 FOR 过程计算，同样可以达到目的。

程序及运行示例如下：

程序 3-7-3

```
TO FSH2 :N
  MAKE "T 0 MAKE "R 0
  FOR "K 1 :N 1 [MAKE "R :R+1/:K
MAKE "T :T+1/ : R]
  PR :T
END

? FSH2 10
4.98675
```

3-8　求N个数的最大公约数及其最小公倍数

编程求任意N(≥2)个正整数的最大公约数及其最小公倍数。

解　求二数最大公约数，可用辗转相减、辗转相除或逐个搜索。若最大公约数是G，则它们的最小公倍数L＝A×B/G。然后用同样的办法求第三个数C与G的最大公约数，及C与L的最小公倍数。如此反复不断，直至所有的数。故，从第三数开始，只要把G给A，C给B，求最大公约数再给G。又只要把L给A，C给B，求最小公倍数等。

主过程GAL有一个输入，以表给出全部数据。子过程GCD求二数最大公约数，用辗转相除法。子过程GBL递归调用本身对全部数据重复工作，直至结束。最后打印出所要求的数。

程序清单及运行示例如下：

程序 3-8

```
TO GAL :D
  CLEARTEXT PR :D
  IF :D = [ ] TOPLEVEL
  MAKE "A FIRST :D
  MAKE "B FIRST BF :D
  MAKE "G GCD :A :B
  MAKE "L :A * :B / :G
  GBL BF BF :D
END
```

```
TO GBL :D
  IF :D = [  ] (PR "GCD= :G) (PR
"LCM= :L) STOP
  MAKE "A :G
  MAKE "B FIRST :D
  MAKE "G GCD :A :B
  MAKE "A :L
  MAKE "L :L * :B / GCD :A :B
  GBL BF :D
END

TO GCD :A :B
  IF REMAINDER :A :B = 0 OP :B
  OP GCD :B REMAINDER :A :B
END

?GAL [25 75 125]
25 75 125
GCD= 25
LCM= 375.

?GAL [63 45 72]

63 45 72
GCD= 9
LCM= 2520.
```

3-9 人口问题

计划生育是基本国策，否则将发生“人口爆炸”！请试编程解决如下的问题。若每年人口的平均增长率分别为3%、2.5%、2%、1.5%、1%、0.5%时，求①到公元2000年时；②到公元2050年时；③再过1000年以后。我国人口的总数及每平方米上人口的密度。人口基数以1985年的11.83亿人为准。全国土地面积约按960万平方公里计算。

解 题目实际很简单。利用公式

$$现有数=基数\times(1+增长率/100)^{n}。$$

问题是如何编程使三个年头、每个年头分别用六个不同的增长率计算出不同的人口数来。

我们编制如下三个过程。主过程NDS以表输入三个年代，递归三次。调用子过程ZZL，该过程以C和N为输入，N为年代，C为增长率，每次减0.5，递归六次。再调用子过程RKS，RKS以年限为终止条件，计算出每个增长率的人口数来。

程序及运行结果如下：

程序 3-9

```
TO NDS :L ; [2000 2050 2985]
  IF :L = [ ] STOP
  MAKE "N (FIRST :L) - 1985
  PR FIRST :L
  ZZL :N 3
  NDS BF :L
```

```
END
TO ZZL :N :C ; (3 2.5 2 1.5 1 .5)
  IF :C < .5 STOP
  MAKE "R 118300
  RKS :N 1
  ZZL :N :C - .5
END
TO RKS :N :I
  IF :I > :N PRINT1 "' ' (PR "C=
:C "R= :R) STOP
  MAKE "R :R * (1 + :C / 100)
  RKS :N :I + 1
END
? NDS [2000 2050 2985]
2000
  C= 3 R= 184307.
  C= 2.5 R= 171333.
  C= 2. R= 159216.
  C= 1.5 R= 147902.
  C= 1. R= 137342.
  C= .5 R= 127489.
2050
    C= 3 R= 807976.
    C= 2.5 R= 588881.
    C= 2. R= 428539.
```

```
    C =  1.5  R =  311364.
    C =  1.  R =  225875.
    C =  .5  R =  163595.
2985
    C =  3  R =  8.13066E17
    C =  2.5  R =  6.26202E15
    C =  2.  R =  4.71059E13
    C =  1.5  R =  3.45857E11
    C =  1.  R =  2.47902E9
    C =  .5  R =  1.73347E7
```

3-10 解一元二次方程

编程求一元二次方程 $ax^2+bx+c=0$ 的根。

解 编制一个解一元二次方程的程序，需要根据 a、b、c 三个系数的情况，加以考虑。为了尽可能地完整些，我们分如下一些情况：

① 当 $a=0$、$b=0$ 时，输出出错，执行停止。

② 当 $a=0$，$b\neq0$ 时，一次方程有解 $x=-c/b$。

③ 当 $c=0$，$a\neq0$ 时，有解 0 和 $-\dfrac{b}{a}$。

④ 当 $\Delta=b^2-4ac\geqslant0$ 时，有一对实根(含重根)。

⑤ 当 $\Delta=b^2-4ac<0$ 时，有一对复根。

前三种情况比较简单，后两种情况比较复杂，形式差异较大。因而分别编写子过程打印输出。

程序由四个过程构成。主过程 FCH 。子过程 SHG、

FUG 及 ABS。子过程分别输出实根、复根及绝对值。主过程名键入并回车后，由键盘输入方程的三个系数 a、b、c，存入变量 L 中。程序然后根据不同的情况进行分支处理。

程序及运行示例如下：

程序 3-10

```
TO FCH
  MAKE "L RQ ; A B C
  MAKE "A FIRST :L
  MAKE "B FIRST BF :L
  MAKE "C LAST :L
  IF ALLOF :A = 0 :B = 0 PR "ERR
STOP
  IF ALLOF :A = 0 NOT :B =0 (PR
"X= (- :C / :B) ) STOP
  IF ALLOF :C = 0 NOT :A = 0 (PR
"X1=0 "X2= - :B / :A) STOP
  MAKE "D :B * :B - 4 * :A * :C
  MAKE "R (- :B / 2 / :A)
  MAKE "S (SQRT (ABS :D) ) / 2 /
  ABS :A
  IF :D < 0 FUG :R :S STOP
  SHG :R :S
END

TO SHG :X :Y
  (PR "X1= :X + :Y)
```

```
  (PR "X2= :X - :Y)
END

TO FUG :X :Y
  (PR "X1= :X "+I :Y)
  (PR "X2= :X "-I :Y)
END

? FCH
X1= -1.
X2= -2.

? FCH
X1= 2.
X2= 1.

? FCH
X1= 3.
X2= 3.

? FCH
X1= 1. +I 2.
X2= 1. -I 2.
```

3-11　母牛生殖问题

有一头母牛，它每年年初生一头小母牛。每头小母牛从第四个年头起，每年年初也生一头小母牛。问在第五十年时共有多少头母牛？试编程求解。

解 1 由题意可知，年代和牛的头数，有如下的对应关系：

牛头数	2	3	4	6	9	13	19	28	…	?
年代数	一	二	三	四	五	六	七	八	…	五十

用 n 表示年数，u_n 表示牛的头数，可得：

$$u_4 = u_3 + u_1 = 4 + 2 = 6$$

$$u_5 = u_4 + u_2 = 6 + 3 = 9$$

$$u_6 = u_5 + u_3 = 9 + 4 = 13$$

$$\vdots$$

即递推公式为

$$u_n = u_{n-1} + u_{n-3} (n \geqslant 4)$$

于是，编程方法可以类似于 2-5 题，同样使用复合字作变量名。前三年牛头数作初值，均已知。从第四年起，牛头数为某两年之和。程序为过程 MNT1，带一个输入，从 4 开始。

程序 3-11-1

```
TO MNT1 :I ;  4
  MAKE "A1 2 MAKE "A2 3 MAKE "A3 4
  PR :A1 PR :A2 PR :A3
  CC: IF :I > 50 STOP
  MAKE WORD "A :I (THING WORD "A :
I - 1 ) + ( THING WORD "A :I - 3 )
  PR THING WORD "A :I
  MAKE "I :I + 1
  GO "CC
END
```

键入 MNT 1　4 回车，则打印出每年牛的头数，直至第五十年(运行结果略)。

解 2　同样基于如上的分析，若以 A、B、C 三个变量分别表示第一年、第二年、第三年母牛的头数，即初值

$$A=2 \quad 第一年$$
$$B=3 \quad 第二年$$
$$C=4 \quad 第三年$$

此后

$$A+C=2+4=6\rightarrow A=6 \quad 第四年$$
$$A+B=6+3=9\rightarrow B=9 \quad 第五年$$
$$B+C=9+4=13\rightarrow C=13 \quad 第六年$$
$$\vdots$$

欲求第五十年牛的头数，应

$$(50-3)/3=15\cdots2$$

即第 16 次的第二个 B 的值。

故 I 从 1 开始到 16 结束。打印出 B，就是答案。

程序及运行结果如下：

程序 3-11-2

```
TO MNT2 :I ; 1
  MAKE "A 2 MAKE "B 3 MAKE "C 4
  CC: IF :I > ( QUOTIENT ( 50 - 3 )
3 ) + 1 PR :B STOP
  MAKE "A :A + :C MAKE "B :A + :B
MAKE "C :B + :C
  MAKE "I :I + 1
  GO "CC
```

```
END
? MNT2 1
262271568
```

3-12 小兔繁殖问题

一个饲养场引进一对刚出生的新品种小兔子。这对兔子从出生的第二个月后开始，每月新生一对小兔子。新生小兔子也都从第二月后开始每月生一对小兔。如果所有兔子都不死去，问到第十二个月时，这个饲养场一共有多少对这样的兔子？

解 由题意可知月份和每个月兔子的对数，有如下的对应关系：

月份	1	2	3	4	5	6…	11	12
对数	1	1	2	3	5	8……		?

第一个月一对小兔，第二个月未生小兔，仍为一对；第三个月开始有新的小兔出生，所以是两对。第四个月只有老兔生小兔，所以是三对，等等。可以看出，从第三个月开始，以后每个月兔子的对数，都是前两个月兔子对数的和。于是，定义过程 XTU 带三 个输入，1 月份兔子数、2 月份兔子数 和月份。即 A 为 1，B 为 1，I 从 1 开始，大于 12 停止。如此打印出每个月份兔子的对数 T1 等。最后一个 T12，就是 12 月时全场兔子的对数。

程序及运行结果如下：

程序 3-12-1

```
TO XTU :A :B :I ; 1 1 1
```

```
  IF :I > 12 STOP
  ( PRINT1 WORD "T :I "= ) PR :A
  ( PRINT1 WORD "T :I + 1 "= ) PR :B
  MAKE "A :A + :B
  MAKE "B :A + :B
  XTU :A :B :I + 2
END

?XTU 1 1 1
T1=1
T2=1
T3=2
T4=3
T5=5
T6=8
T7=13
T8=21
T9=34
T10=55
T11=89
T12=144
```

还可建立过程 XT2，只打印出 12 月份时所有兔子的总数 T。

程序 3-12-2

```
TO XT2 :A :B :I ; 1 1 1
  IF :I > 5 ( PR "T= :B ) STOP
```

```
    MAKE "A :A + :B
    MAKE "B :A + :B
    XT2 :A :B :I + 1
  END

  ?XT2 1 1 1
  T= 144
```

3-13　求一个数列的前N项

有一个数列，它的前3项是0、0、1，从第4项开始，其后每一项都是它相邻前3项之和。编程打印出这个数列的前20项。

解　数列前三项已知，从第四项起，后边每一项都是它前面三项之和，即

$$A4 = A3 + A2 + A1$$
$$A5 = A4 + A3 + A2$$
$$A6 = A5 + A4 + A3$$
$$\vdots$$

若以复合字 WORD "A :I 为变量名，I 从 4 到 20，即可表示 A4，A5，…，A20 各项。于是，可用一个表达式给各项赋值。

定义过程 SLS，带一个输入，从4开始。利用 GO 命令设置一个循环模块，当 I 大于 20 后结束。

清单如下。只要键入 SLS 4 回车，即可打印出数列的前 20 项。屏幕上数列的前 20 项是按一行一行打印的。

程序 3-13

```
TO SLS :I ;  4
  MAKE "A1 0 MAKE "A2 0 MAKE "A3 1
  PR :A1 PR :A2 PR :A3
  CC: IF :I > 20 STOP
  MAKE WORD "A :I ( THING WORD "A
:I - 1 ) + ( THING WORD "A :I - 2 )
+ ( THING WORD "A :I - 3 )
  PR THING WORD "A :I
  MAKE "I :I + 1
  GO "CC
END

?SLS 4
0 0 1 1 2 4 7 13 24 44 81 149 274
504 927 1705 3136 5768 10609 19513
```

3-14 求式中字母代表的值①

编程求出下式中字母所表示的数字。

$$\frac{AHHAAH}{JDKE}=HA$$

解 左边六个字母，一般见此题后，可能想用六重循环或递归调用穷举可能的所有情况。但这太费事。只要将原式变形为

$$\frac{AHHAAH}{HA}=JDKE$$

那么，只要对 A、H 用二重循环，从 1 到 9 把可能代表的数字代入上式左边计算然后判断它是否为一个四位数即可。

程序由两个过程组成，AA 为主过程。HH 为子过程。主过程带一个输入，从 1 开始。程序及运行结果如下：

程序 3-14

```
TO AA :A ;  1-9
  IF :A > 9 STOP
  HH 1
  AA :A + 1
END

TO HH :H ;  1-9
  IF :H > 9 STOP
  MAKE "X 100110 * :A + 11001 * :H
  MAKE "X :X / ( 10 * :H + :A )
  IF NOT ROUND :X = :X HH :H + 1 STOP
  IF :X < 1000 HH :H + 1 STOP
  IF :X > 9999 HH :H + 1 STOP
  (PRINT1 :A :H :H :A :A :H "/ :X "= :H
:A ) PR "
  HH :H + 1
END

?AA 1
377337/5169. = 73
```

3-15　求式中字母代表的值②

编程求出下式中字母所代表的数字。

$$[A\times(B3+C)]^2=8DE9$$

解　式中五个字母，可用穷举法把所有可能的数字组合代入后加以判断，但这样运行太浪费时间。只要设置三个递归过程，构成三重循环，求出 A、B、C 所代表的数字后代入原式左边，再判断是否满足下列三个条件。(1)是否为四位数，(2)最高位是否为 8。(3)最低位是否为 9 即可。

程序由三个过程构成，其中主过程 BW 确定百位数 A，过程 SW 及 GW 是子过程，分别确定十位数 B 和个位数 C。

程序及运行结果如下。可以看出，满足该式 A、B、C、DE 的值有两组。

程序 3-15

```
TO BW :A ;  1-9
  IF :A > 9 STOP
  SW 1
  BW :A + 1
END

TO SW :B ;  1-9
  IF :B > 9 STOP
  GW 0
  SW :B + 1
END
```

```
TO  GW :C ;  0-9
  IF :C > 9 STOP
  MAKE "X :A * ( 10 * :B + :C + 3 )
  MAKE "X :X * :X
  IF ANYOF :X < 8009 :X > 8999 GW :C
+1 STOP
  IF NOT REMAINDER :X - 9 10 = 0 GW
:C + 1 STOP
  (PRINT1 "'[' :A "*( :B 3 "+ :C "')]'
CHAR 94 "2= :X ) PR "
  GW :C + 1
END
```

```
?BW 1
[1*(93+0)]^2=8649
[3*(23+8)]^2=8649
```

四 有趣的数

4-1 分 316 为二正数之和

把 316 这个数表示为两个正数加数的和，使其中一个加数能被 13 整除，而另一个加数能被 11 整除。编程求出所有这样的正数加数。

解 能被 13 整除的数，一定是 13 的整数倍。设为 N，则 N 可以从 1 到 25。每次加 1，判断 316－(N×13)能否被 11 整除。是则打印，否则继续。

程序清单及运行结果如下。由运行结果可知有两组加数。

程序 4-1

```
TO CCC :N
  IF :N > 25 STOP
  IF REMAINDER 316 - :N * 13 11 = 0
  ( PR :N * 13 "' ' 316 - :N * 13 )
  CCC :N + 1
END

?CCC 1
52     264
195     121
```

4-2 求N!能被M的最高多少次方整除

编程求任意数N!能被一个质数M的最高多少次方整除。如求17!能被2的最高多少次方整除。

解 先算出N!，再用M去除，记下整除的次数，即为所求。但是，如此慢而且麻烦，有时甚至不可能。如数字太大，阶乘内存溢出。根据阶乘的连乘特点，只要对每一个因数进行考查，用M去除从1到N的各个数，凡能整除的总次数，就是所要求的最高次方数。

程序由三个过程构成，主过程SQC，带两个输入，分别为N和M。如若要求17!，能被质数2的最高多少次方整除，只要键入SQC 17 2回车，即得结果。

程序及运行结果示例如下：

程序 4-2

```
TO SQC :N :C
  MAKE "K 0
  SCF 1 :N
  ( PRINT1 :N "!/ )
  ( PRINT1 :C CHAR 94 :K )
  PR "' '
END

TO SCF :I :N
  IF :I > :N STOP
  MAKE "M :I
  RCS :M
```

```
  SCF :I + 1 :N
END

TO RCS :J
  IF :J < :C STOP
  IF REMAINDER :J :C = 0 MAKE "K :K
+ 1 RCS :J / :C
END

?SQC 33 2
33!/2^31

?SQC 500 7
500!/7^82

?SQC 3 2
3!/2^1

?SQC 2 3
2!/3^0
```

4-3 尾巴为 6 的数

有一个多位数，个位为 6。若将 6 去掉前移至首位，则所得的新数是原数的 4 倍。求这个最小的多位数是多少。

解法 1 个位是 6 的多位数，从 16 开始。每次递加 10 穷举累试。一定能找到所求的。为此设变量 N，用复合字命令将 6 前移到首位，送变量 M。判断 M 是否为 N 的 4 倍。是则打印结束，否则加 10 再试。

建立两个过程，主过程为 WB6，子过程为 WB61。程

序及运行结果如下：

程序 4-3-1

```
TO WB6 :N; 16
  MAKE "M WORD 6 BL :N
  IF :M=:N*4 (PR :N "----- :M) STOP
  WB6 :N + 10
END

TO WB61 :N ; 16
  AA: MAKE "M WORD 6 BL :N
  IF :M=4 * :N (PR :N"----- :M) STOP
  MAKE "N :N+10 GO "AA
END

?WB61 16
153846 ----- 615384
```

解法 2 由题可知，假设多位数为 A…ST6，则应有如下算式

$$
\begin{array}{r}
A\cdots ST6 \\
\times)\qquad 4 \\
\hline
6A\cdots ST
\end{array}
$$

但 6×4＝24，因此 T＝4；4×4 再加刚才的进位 2，得 18，因此 S＝8；…。这样由右向左，一直继续下去。直到某一位出现 6 且不再有进位为止。这样，所求的多位数便确定了。

为此，建立两个过程，主过程 SHI，带两个输入，M 为

6，N为4。设置三个变量，Z放最后结果，G放每次相乘的个位，S放每次相乘的十位(也即进位)。

程序及运行结果如下：

程序 4-3-2

```
TO SHI :M :N; 6 4
  MAKE "Z 6
  MAKE "G 0
  MAKE "S 0
  CHI :M :N
END

TO CHI :M :N
  MAKE "C :M * :N
  IF BF :C =" PR :Z TOPLEVEL
  MAKE "G :S + LAST :C
  MAKE "S FIRST :C
  MAKE "Z (WORD :G :Z)
  CHI :G :N
END

?SHI 6 4
153846
```

可见所求的数为153846，与前一程序结果相同。

4-4 平方中含4的数

编写一个程序，在100以内的自然数中寻找出这样的数，

①它的平方是至少有两个 4 的数。如 22^2 是 484。②它的平方是至少有两个连续的 4 的数。如12^2 是 144。

解 ①对 1 到 100 的自然数，先求它的平方，对平方的各个数位进行判断。如果是 4，就给记数变量 K 加 1 。最后打印出 K 是 2 以上的 N 及其平方。如此继续，直到 100。

程序两个过程。PFS 是主过程，第一个输入，即由1开始的自然数。子过程 PIK,对所给自然数的平方的各位数字进行检测判断。

程序及运行结果如下：

程序 4-4-1

```
TO PIK :I
  IF :I = " STOP
  IF FIRST :I = 4 MAKE "K :K + 1
  PIK BF :I
END

TO PFS :N; 1-100
  IF :N > 100 STOP
  MAKE "K 0
  MAKE "M :N * :N
  PIK :M
  IF NOT :K < 2 (PR :N "'      ' :M)
  PFS :N + 1
END

?PFS 1
12          144
```

21	441
22	484
38	1444
62	3844
67	4489
68	4624
88	7744
92	8464

② 与上类似，不同的是在求出平方数之后，要对连续的两位分别判断，看其是否为44。如果有则给记数变量加1。每个数判断以后，看 K 值是否等于或大于 1，是即打印出N值及其平方。程序类似，主过程为 PFS1，子过程为PIK1。程序及运行结果如下：

程序 4-4-2

```
TO PIK1 :M
  IF BF :M = " STOP
  MAKE "S WORD FIRST :M FIRST BF :M
  IF :S = 44 MAKE "K :K + 1
  PIK1 BF :M
END

TO PFS1 :N; 1-100
  IF :N > 100 STOP
  MAKE "K 0
  MAKE "M :N * :N
  PIK1 :M
```

```
  IF NOT :K < 1 (PR :N "'      ' :M)
  PFS1 :N + 1
END

?PFS1 1
12          144
21          441
38          1444
62          3844
67          4489
88          7744
```

可以看出，在①的结果中，包含了②的结果。前者虽有含两个4的，但若不连续，则便不属于后者。

4-5 “水仙花”数

数153有这样的性质，它是一个三位数，而它的各位数字的立方和正好与它相等。把这样的三位数，称为“水仙花”数。编程求出三位数中的所有“水仙花”数。

解法1 三位数是从100到999的数。对每一个数，分别分离出各位数字，求它们的立方和，若立方和等于原数的，打印输出。不相等则继续下去直至全部。程序只一个过程。A、B、C分别表示百、十及个位数字。

程序如下：

程序 4-5-1

```
TO SXH :N; 100-999
```

```
  IF :N > 999 STOP
  MAKE "A FIRST :N
  MAKE "B FIRST BF :N
  MAKE "C LAST :N
  IF :A * :A * :A + :B * :B * :B +
:C * :C * :C = :N PR :N
  SXH :N + 1
END
```

解法 2 建立一个求乘方运算的子函数过程 POW，分离出百、十、个位数之后直接调用 POW 求它们的三次方，然后存放在 A、B、C 中。主过程 SXH1。

程序及运行结果如下。可知三位数中有四个水仙花数。

程序 4-5-2

```
TO SXH1 :N; 100-999
  IF :N > 999 STOP
  MAKE "A POW FIRST :N 3
  MAKE "B POW FIRST BF :N 3
  MAKE "C POW LAST :N 3
  IF :A + :B + :C = :N PR :N
  SXH1 :N + 1
END

TO POW :A :E
  IF :E =0 OP 1
  IF :E = 1 OP :A
  OP :A * POW :A :E - 1
```

```
END
?SXH 150
153
370
371
407
```

4-6 和位幂数

数 512 有这样的性质，它是三位数，而各位数字之和的三次方正好是原数。即

$$512=(5+1+2)^3$$

编程求出两位数至四位数中所有具有这种性质的数。它的各位数字之和的 m 次方正好是原数，这个 m 是原数的位数。

解法 1 用最基本的办法，一个一个搜索。先求所给数的位数，存变量 M；再求各位数字之和存变量 H；再求 H 的M 次方存变量 C；最后检查施行如上运算后的 C 与输入数是否相等，是则打印输出，否则继续下一个。

程序由五个过程组成。HMS 为主过程，带一个输入数。调用其它子过程分别完成各项工作。其中 LEN 过程计数位数。ADD 过程求各位数字之和。POW 过程是一个简化的乘方运算过程。PRT 过程用来按一定的格式打印结果。

程序及运行结果如下：

程序 4-6-1

```
TO HMS :S
  MAKE "M LEN :S
```

```
  MAKE "H ADD :S
  MAKE "C POW :H :M
  IF :C = :S (PRINT1 :S "=(  ) PRT
:S PR :M
  HMS :S + 1
END

TO LEN :N
  IF :N = "" OP 0
  OP 1 + LEN BF :N
END

TO ADD :N
  IF :N = "" OP 0
  OP ( FIRST :N )  + ADD BF :N
END

TO POW :A :E
  IF :E = 0 OP 1
  IF :E = 1 OP :A
  OP :A * POW :A :E - 1
END

TO PRT :N
  IF BF :N = "" ( PRINT1 :N ") CHAR 94)
STOP
  ( PRINT1 FIRST :N "+ )
  PRT BF :N
```

```
END

?HMS 10
81 = (8 + 1)^2
512 = (5 + 1 + 2)^3
2401 = (2 + 4 + 0 + 1)^4
```

以上程序运行之后不能自动终止，可用 CTRL-G操作中断，或在主过程 HMS 第一行命令后增加 IF:M>4 STOP。

逐个检查，速度较慢。改用下面的方法，则十分迅速。

解法 2 我们知道，10 的 2 次方是 100，为三位数。10 的 3 次方是 1000，为四位数。10 的 4 次方是 10000，为五位数。因此，如果两位数至四位数中它的各位数字之和的“位数”次方要正好是原数，那么，各位数字之和必然小于或等于 9 。又知道，一个数的 2 次方要是二位数，该数一定大于或等于 4；同理，3 次方的三位数，一定大于或等于 5；4 次方的四位数，一定大于或等于 6 。可以看出，各位数字之和与位数之间有这样的关系：$2+M\leqslant N\leqslant 9$。（N 为和，M 为位数）。

因此，我们建立主过程 HMS1，它带一个输入，表示位数从 2 至 4；子过程 HMS2，它的输入从位数加 2 至 9。再调用子过程 ADD、POW 及 PRT。先求 N 的 M 次方送S，再求 S 各位数字之和送 H，判断 H 与输入 N 是否相等。相等则打印输出。否则继续下一个。

程序及运行结果如下。其余子过程同解 1 。

程序 4-6-2

```
TO HMS1 :M; 2-4
  IF :M > 4 STOP
```

```
  HMS2 :M + 2
  HMS1 :M + 1
END

TO HMS2 :N; :M+2-9
  IF :N > 9 STOP
  MAKE "S POW :N :M
  MAKE "H ADD :S
  IF :H = :N ( PRINT1 :S "=(  ) PRT
:S PR :M
  HMS2 :N + 1
END

?HMS1 2
81=(8+1)^2
512=(5+1+2)^3
2401=(2+4+0+1)^4
```

4-7 同 构 数

打印1000以内的同构数。所谓同构数，是指一个自然数N是它本身平方后的尾数，就称N为同构数。例如5的平方是25，5是25的尾数，5称为同构数。

解法1 采用递归搜索，从1开始至1000结束。设置变量I记数位数，M存放平方。然后测试数是否与尾数相同。是，则打印，N+1继续；不是，直接N+1继续。计字长调用LEN过程，比较尾数调用RIGHT$过程。TGS1是主过程。程序及运行结果如下：

程序 4-7-1

```
TO TGS1 :N ; 1-1000
  IF :N > 1000 STOP
  MAKE "I LEN :N
  MAKE "M :N * :N
  IF :N = RIGHT$ :M :I ( PR :N "'
:M ) TGS1 :N + 1 ELSE TGS1 :N + 1
END
```

```
?TGS1 1
1          1
5          25
6          36
25          625
76          5776
376          141376
625          390625
```

可知，1000 之内共有七个同构数。

解法 2 基本思想同 1。比较尾数时不用字函数，而是调用 POW 过程，将平方数用 10 的 I 次方除，求余数与 N 比较。该法调用子过程 LEN 及 POW，主过程 TGS2。

程序如下：

程序 4-7-2

```
TO TGS2 :N
  IF :N > 1000 STOP
  MAKE "I LEN :N
```

```
  IF :N = REMAINDER :N * :N POW 10
:I ( PR :N "'     ' :N * :N ) TGS2 :N+ 1
ELSE TGS2 :N + 1
END
```

4-8 回 文 数

编程求不超过八位数范围内既是完全平方数又是回文数的所有数。所谓完全平方数，是指它的正平方根是自然数；所谓回文数，指的是两位以上的数其数字前后对称。如121，4224 等。

解 首先确定范围。平方是两位的自然数最小是 4，在所要求范围内的数，不能大于 9999。于是我们可将 4～9999 间的数，逐一平方，再检查它是否左右对称。

法 1 由 HW1、H1 及 MID$、LEN 四个过程构成一个程序。程序如下：

程序 4-8-1

```
TO HW1 :N ; 11-9999
  IF :N > 9999 STOP
  MAKE "A :N * :N
  MAKE "L LEN :A
  MAKE "S 0
  H1 :A 1
  HW1 :N + 1
END

TO H1 :W :J
```

```
  IF :J > :L / 2 ( PR :W "= :N "* :N )
STOP
  IF MID$ :W :J 1 = MID$ :W :L + 1
- :J 1 H1 :W :J + 1
END
```

法 2 对求出的每一个平方数，使用 LOGO 原语命令 FIRST 与 LAST 分别取出前一个和最后一个比较，判定是否回文。程序由两个过程组成。HW2 为主过程，H2 为子过程。程序如下：

程序 4-8-2

```
TO HW2 :N ; 11-9999
  IF :N > 9999 STOP
  MAKE "A :N * :N
  H2 :A
  HW2 :N + 1
END

TO H2 :W
  IF ANYOF :W = "" BF :W = "" ( PR
:A "= :N "* :N ) STOP
  IF FIRST :W = LAST :W H2 BF BL :W
END
```

法 3 建立如下三个过程，主过程为 HW3，两个子过程 H3 和 R3。这里判断回文的方法，是先将判断对象全部颠倒过来，看与原来是否相同。程序及整个运行结果如下：

程序 4-8-3

```
TO HW3 :N ; 11-9999
  IF :N > 9999 STOP
  MAKE "A :N * :N
  TEST H3 :A
  IFT ( PR :A "= :N "* :N ) HW3 :N
+ 1
  IFF HW3 :N + 1
END

TO H3 :K
  OP ( :K = R3 :K)
END

TO R3 :I
  IF :I = "" OP ""
  OP WORD LAST :I R3 BL :I
END

?HW3 11
121 = 11 * 11
484 = 22 * 22
676 = 26 * 26
10201 = 101 * 101
12321 = 111 * 111
14641 = 121 * 121
40804 = 202 * 202
44944 = 212 * 212
```

```
69696 = 264 * 264
94249 = 307 * 307
698896 = 836 * 836
1002001 = 1001 * 1001
1234321 = 1111 * 1111
4008004 = 2002 * 2002
5221225 = 2285 * 2285
6948496 = 2636 * 2636
```

该程序运行时间较长。但从运行结果可以看出，终止条件用9999，机器做许多无用功。其实可将终止条件改为3000，同样可得如上的结果，却为你节约了很多宝贵的时间。

4-9 有趣的整数

有这样一些数，用 2 至 9 这 8 个数字相乘之后，所得的数的各位数字之和，与原来各位数字之和都相等。编程求出某个指定范围具有这种特性的所有数。

例如，

$$18\times2=36 \qquad 3+6=1+8=9$$

$$18\times3=54 \qquad 5+4=1+8=9$$

$$18\times4=72 \qquad 7+2=1+8=9$$

$$18\times5=90 \qquad 9+0=1+8=9$$

$$\vdots \qquad\qquad \vdots$$

$$18\times9=162 \qquad 1+6+2=1+8=9$$

解 数学上可以证明，这一类数必须是 9 的倍数。为了减少测试次数，提高运行速度，程序输入初值从 9 开始，且以后每次加 9 进行测试。

程序两个过程。YQS 为主过程，QHE 为子过程。子过程用于求一个数各位数字之和，存变量 H。主过程 YQS 有两个输入，第一个是需要检查的数 S，从 9 开始。第二个 M，指限定的范围。设置量 SH 为输入数本身各位数字之和。JH 为乘积各位数字之和。BB 循环使 N 从 2 到 9，分别去乘输入数。若满足条件，则打印输出，有一个不满足条件，即加 9 递归，对另一个数重新开始。

程序及运行示例如下(求出了 500 内具有这种性质的 11 个数)：

程序 4-9

```
TO YQS :S :M ; 9-M
  IF :S > :M STOP
  MAKE "H 0
  QHE :S MAKE "SH :H
  MAKE "N 2
  BB: IF :N> 9 PR :S YQS :S + 9 :M STOP
  MAKE "J :S * :N
  MAKE "H 0
  QHE :J MAKE "JH :H
  IF :SH = :JH MAKE "N :N + 1 GO "BB
  YQS :S + 9 :M
END

TO QHE :S
  IF :S = " STOP
  MAKE "H :H + FIRST :S
```

```
  QHE BF :S
END
?YQS 9 500
9
18
45
90
99
180
198
297
396
450
495
```

4-10　不怕撕裂的数

有这样一些数，它们的特征是：按数位把它们对半分成两个数，然后将这二数相加，和再平方。结果还是原数。如

$$81=(8+1)^2=9^2=81$$

$$3025=(30+25)^2=55^2=3025$$

试编程求出 8 位数内具有这种性质的所有数。

解　对所给的输入数，先求它的平方，并求平方数的位数。然后按数位左、右一分为二，求和并平方。最后检查结果是否与原数相等。相等则打印，接着下一个；不相等则直接去试下一个。

建立主过程LS1。它调用字函数过程LEN、LEFT$及RIGHT$为其子过程。

同时，要能前后分为两段的数，位数是奇数的当然不行。所以过程对位数的奇偶事先加以检查。

程序如下：

程序 4-10-1

```
TO LS1 :N; 0-9999
  IF :N > 9999 STOP
  MAKE "A :N* :N
  MAKE "L LEN :A
  IF NOT REMAINDER :L 2=0 LS1 :N+1
  MAKE "B (LEFT$ :A :L/2) + (RIGHT$
:A :L/2)
  MAKE "B :B* :B
  IF :B= :A PR :B
  LS1 :N+1
END
```

应当说明，运行此程序，不能一次连续求出八位内具有此性质的全部数。应分别用不同的输入运行几次。如键入LS1 9回车，待光标出现后再键入LS1 700回车，待光标出现后再键入LS1 4950回车等，才可求出八位内11个不怕撕裂的数。

程序若改为如下的程序LS2，则只要输入从9开始，便可一次得出全部结果。

程序及运算结果如下：

程序 4-10-2

```
TO LS2 :N; 0-9999
  AA: IF :N > 9999 STOP
  MAKE "A :N* :N
  MAKE "L LEN :A
  IF NOT REMAINDER :L 2=0 MAKE "N
:N+1 GO "AA
  MAKE "B (LEFT$ :A :L / 2) +
(RIGHT$ :A :L/2)
  MAKE "B :B* :B
  IF :B= :A PR :B
  MAKE "N :N+1 GO "AA
END
```

```
 ? LS2 9
 81
 2025
 3025
 9801
 494209
 998001
 24502500
 25502500
 52881984
 60481729
 99980001
```

程序用截尾取整函数代替求余函数，另调用 MID$ 及 PICK 等常用的过程为子过程，效果是相同的。

4-11 求质数

编一个程序，能根据要求求出指定范围内的所有质数。

解 质数，也叫素数。是只能被 1 及其本身整除的自然数。要判断某数 n 是否为质数，只要从定义出发，用 2 至 $\sqrt{n}$ 间的每一个整数去除它，如果都不能整除，即为质数。

法 1 建立两个过程。主过程 ZS，两个输入，2 及要求质数的范围。子过程 ZS1，它也带两个输入，一个为 2，另一个为被判断的数。ZS1 由 ZS 调用，对于每一个整数，分别用 2 到 $\sqrt{n}$ 间的每一个整数去除。只要有一个可整除，就结束返回主过程加 1 判断下一个。否则，直到 $\sqrt{n}$ 未有可整除的，则打印该数，再返回主过程取下一个。

程序及运行结果如下：

程序 4-11-1

```
TO ZS :I :B; 2 RAN
  IF :I > :B PR "' ' STOP
  ZS1 2 :I
  ZS :I+1 :B
END

TO ZS1 :J :K
  IF :J>.1+SQRT :K (PRINT1 :K "' ') STOP
  IF REMAINDER :I :J=0 STOP
  ZS1 :J+1 :K
```

```
END

  ?ZS 2 100
  2 3 5 7 11 13 17 19 23 29
  31 37 41 43 47 53 59 61
  67 71 73 79 83 89 97
```

运行结果求出了 100 以内的 25 个质数。

法 2 质数除 2 之外，均为奇数。若从 3 开始，每次加 2，只在奇数中加以判断，无疑可减少递归次数，大大加快程序运行速度。

为此，我们建立三个过程。主过程 ZS2，带两个输入，3及指定的范围。它先打印出 2 来，然后再调用子过程 ZS21。ZS21 从 3 开始，到指定的范围内，每次加 2。调用子过程 ZS22。子过程 ZS22 则对每一个要检测的数，用 3 到该数的平方根间的数，每次加 2，进行检查。只要有一个可整除，即返回 ZS21 取另一个数。否则直至结束打印输出，再返回取下一个。

此程序比法 1 程序速度快多了。

程序如下(能得到与法 1 同样的结果)：

程序 4-11-2

```
TO ZS2 :I :B; 3 RAN
  (PRINT1 2 "' ')
  ZS21 :I :B
END

TO ZS21 :I :B; 3 RAN
  IF :I > :B PR"' ' STOP
```

```
  ZS22 3 :I
  ZS21 :I+2 :B
END

TO ZS22 :J :K
  IF :J > .1+SQRT :K (PRINT1 :K
"' ') STOP
  IF REMAINDER :I :J=0 STOP
  ZS22 :J+2 :K
END
```

4-12 分解质因数

编一个程序，能对输入的一个正整数或负整数，求出它的所有素数因数，并打印出这些素因数连乘积的形式。

解 首先区分正负数，若是负数，前面先打印出符号，并取其相反数进行分解。

要求一个数的因数，我们从2开始，每次加1，逐个试除，直到该数的平方根。能整除即因数，不能整除，便不是因数。至于同一个因数，试除之后，将相除结果作被除数再试。

过程SCD如下。N为输入。求出一个因数之后，将相除结果又送N中。Y为试除因数，从2开始。Z存放刚一开始所求数N的平方根。

程序及运行示例如下：

程序 4-12

```
TO SCD :N
```

```
  PRINT1 :N
  IF :N < 0 PRINT1 "= - MAKE "N (- :N)
ELSE PRINT1 "=
  MAKE "Y 2
  MAKE "Z SQRT :N
  AA: IF :Y > :Z PR :N STOP
  IF NOT REMAINDER :N :Y=0 MAKE "Y
:Y+1 GO "AA
  MAKE "N :N / :Y
  IF :N < :Y PR :Y STOP ELSE (PRINT1
:Y "* ) GO "AA
END

?SCD 36
36=2*2*3*3
?SCD 98765
98765=5*19753.
?SCD 19753
19753=19753
?SCD -864
-864=-2*2*2*2*2*3*3*3
```

可以看出，如果输入是一个质数，它将原样印出。

4-13 史密斯数

这是一种新近发现尚鲜为人知的数。它的特点是，本身

是合数，它所有数位上的数字之和等于它全部因子的数字之和。比如22，2+2=4。22=2×11，而2+1+1=4。编程寻找某个范围内的史密斯数。

解 据说史密斯数的发现纯粹是一种偶然的机会。美国数学家阿·威兰斯基与他的朋友史密斯打电话的时候，发现他的电话号码4937775是一个怪数，首先它是合数，其因子为4937775=3×5×5×65837，而

$$4+9+3+7+7+7+5=42$$

$$3+5+5+6+5+8+3+7=42$$

于是数学家潜心钻研，发现具有此类性质的数还有许多。如最小的是4，其次是22，27…。人们便把这类数称作史密斯数。

我们编制五个过程，主过程SMS，设置变量S1存放各数位数字之和，S2存放各因子数位数字之和。变量L为存放各因子的表。子过程SUW求数字之和，SUL求表中各数字之和。SCD求一个数的因子。LEN是常用过程，求表长。主过程有两个输入，即从几到几范围内的史密斯数。

下面是程序及从4到100及从4到500两个范围内史密斯数的运行结果：

程序 4-13

```
TO SMS :I :M
  IF :I > :M PR "' ' STOP
  MAKE "Z :I
  MAKE "L [ ]
  MAKE "S1 SUW :I
  MAKE "L SCD 2
```

```
  IF ( LEN :L ) < 2 SMS :I + 1 :M S
TOP
  MAKE "S2 SUL :L
  IF :S1= :S2 ( PRINT1 :I "' ' )
  SMS :I+1 :M
END

TO SCD :Y
  IF :Y > SQRT :I OP LPUT :Z :L
  AA: TEST REMAINDER :Z :Y=0
  IFF OP SCD :Y+1
  IFT MAKE "L LPUT :Y :L MAKE "Z :Z
/ :Y
  IF :Z < :Y OP :L
  GO "AA
END

TO SUW :W
  IF :W= "'' OP 0
  IF LAST :W= ". OP SUW BL :W
  OP ( FIRST :W ) +SUW BF :W
END

TO SUL :L
  IF :L=[ ] OP 0
  OP ( SUW FIRST :L ) +SUL BF :L
END
```

```
TO LEN :L
  IF :L=[ ] OP 0
  OP 1+LEN BF :L
END

?SMS 4 100
4 22 27 58 85 94

?SMS 4 500
4 22 27 58 85 94 121 166 202
265 274 319 346 355 378 382
391 438 454 483
```

人们研究发现，1～10000 间共有 376个史密斯数。读者可以试验，看是否确有这么多。

4-14 卡布列克运算

对于给定的各位数字完全不同的一个三位数，先把它的各位数字按大小排序，得到一个大数。再把各位数字按小大排列，得到一个小数。求两者的差。对得到的差，又施行如上的运算。这样经过有限次之后，总能得到 495 这个数。同样，对于各位数字完全不同的四位数，施行如上的运算，总能得到 6174 这个数。把这种有趣现象，叫作“卡布列克运算”。编程验证之。

解 对于给定的数，按要求需要从大到小排序，又需要从小到大排序。将两次排序的结果相减，判断差是否 495，或 6174。这当然要区别三位和四位。是则停止，不是则递

归，又从头开始。

建立如下八个过程。KBL为主过程。带一个输入。将输入数排序后，大数放变量D中，小数放变量X中，其二者之差放变量C中。子过程PAD、PBD从大到小排序。子过程PAX、PBX从小到大排序。另外的子过程LEN计字长，PIK输出指定字符，ERA删去指定字符。

程序及运行示例如下：

程序 4-14

```
TO KBL :W
  ( PRINT1 :W "' ' )
  MAKE "L LEN :W
  MAKE "D "''
  PAD :L :W
  ( PRINT1 :D "- )
  MAKE "X "''
  PAX :L :W
  ( PRINT1 :X "= )
  MAKE "C :D- :X PR :C
  IF ALLOF :L= 3 :C= 495 STOP
  IF ALLOF :L= 4 :C= 6174 STOP
  KBL :C
END

TO LEN :N
  IF :N= "' ' OP 0
  OP 1+LEN BF :N
```

```
END

TO PAD :T :W
  IF :W= "' ' STOP
  MAKE "M FIRST :W
  MAKE "I 1
  PBD 1 :W
  MAKE "D WORD :D :P
  PAD :T-1 ERA :I :W
END

TO PBD :K :W
  IF :K > :T MAKE "P :M STOP
  MAKE "R PIK :K :W
  IF :M <:R MAKE "M :R MAKE "I :K
PBD :K+1 :W ELSE PBD :K+1 :W
END

TO PAX :S :W
  IF :W= "' ' STOP
  MAKE "H FIRST :W
  MAKE "V 1
  PBX 1 :W
  MAKE "X WORD :X :Q
  PAX :S-1 ERA :V :W
END

TO PBX :K :W
```

```
  IF :K > :S MAKE "Q :H STOP
  MAKE "T PIK :K :W
  IF :H > :T MAKE "H :T MAKE "V :K
PBX :K+1 :W ELSE PBX :K+1 :W
END

TO PIK :J :W
  IF :J=1 OP FIRST :W
  OP PIK :J-1 BF :W
END

TO ERA :N :W
  IF :W= "" STOP
  IF :N= 1 OP BF :W
  OP WORD FIRST :W ERA :N-1 BF :W
END

?KBL 691
691      961-169=792
792      972-279=693
693      963-369=594
594      954-459=495

?KBL 814
814      841-148=693
693      963-369=594
594      954-459=495

?KBL 2895
```

```
2895      9852 - 2589 = 7263
7263      7632 - 2367 = 5265
5265      6552 - 2556 = 3996
3996      9963 - 3699 = 6264
6264      6642 - 2466 = 4176
4176      7641 - 1467 = 6174

?KBL 5174
5174      7541 - 1457 = 6084
6084      8640 - 0468 = 8172
8172      8721 - 1278 = 7443
7443      7443 - 3447 = 3996
3996      9963 - 3699 = 6264
6264      6642 - 2466 = 4176
4176      7641 - 1467 = 6174
```

说明 ①本程序可通用，对于输入为二位到九位（在CEC-I 机上）的数，该程序都可运行。不过必要时应用 CTRL-G 中断。感兴趣的读者，可试试五位数、六位数……有何现象。

② 对于通常概念下的三位数（四位数）及减法运算，“卡布列克”运算要求各位上的数字全不相同，即有如上所述的有趣现象。我国数学家陈景润先生在《初等数论》（科学技术出版社 1984 年）中，把它称作有趣的减法，并从理论上证明了出现此种现象最多所需的步数。

如果将三位数（或四位数）改为三个数码（或四个数码），经排序后相减，除沿用一般概念下的相减外，凡是首位不够减时，均以零补足数码位数。即原三位数码补足三位，原四

位数码补足四位。这样，“卡布列克”运算的范围将更加宽广。那么，它将对任一个不全相同的三位数码(或四位数码)都适用。例如：010，排序为100和001，相减应得099，…，最后总能得495这个数。应当注意的是这儿的010不是一个三位数，而是三个不全相同的数码。

对此，我们只要将原程序主过程第九行改为MAKE"C :D- :X IF LEN :C < :L MAKE "C WORD "0 :C，加第十行为PR :C，余不变。且运行时以字形式输入各数码。

五　古算趣题

5-1　鸡兔同笼问题

试编程求解鸡兔同笼问题。即已知鸡和兔的总头数和总脚数，求鸡和兔各有几只。

解法 1　若已知鸡兔总头数为 T，总脚数为 J。求鸡兔各有几只？设鸡有 x 只，兔有 y 只，则可列方程组

$$\begin{cases} x + y = T \\ 2x + 4y = J \end{cases}$$

所以

$$\begin{cases} x = \dfrac{4T - J}{2} \\ y = \dfrac{J - 2T}{2} \end{cases}$$

用赋值命令及以上公式直接进行运算。

程序只有一个过程 JT1。它带两个输入，分别为总头数和总脚数。变量 X 表鸡的只数，变量 Y 表兔的只数。

程序及运行示例如下：

程序 5-1-1

```
TO JT1 :T :J
  MAKE "X ( 4* :T- :J ) / 2
  MAKE "Y ( :J-2 *:T ) / 2
   ( PR "X=  :X )
   ( PR "Y=  :Y )
```

```
END

?JT1 30 82
X= 19.
Y= 11.

?JT1 30 72
X= 24.
Y= 6.
```

解法 2 从总头数开始，假设全是鸡或者兔，让数量逐次减 1。而另一变量逐次加 1，然后测试总脚数关系式是否满足。若满足即打印输出，否则继续加 1、减 1 测试。这样全部工作由机器进行，体现重复快速的特点。过程设三个输入，分别是总头数、总脚数及一个 0 开始的数。它在过程中实际是兔的只数。

程序及运行示例如下：

程序 5-1-2

```
TO JT2 :T :J :Y
  MAKE "S 2 * :T+4 * :Y
  IF :S= :J ( PR "X= :T ) ( PR "Y=
:Y ) STOP
  JT2 :T-1 :J :Y+1
END

?JT2 30 72 0
X= 24
Y= 6
```

5-2　百马百瓦问题

用现在的话说，意思是有 100 匹马驮 100 箱货。规定大马一匹驮 3 箱，中马一匹驮 2 箱，小马两匹合驮 1 箱。要求 100 匹马正好驮完 100 箱货，且不许有剩余及驮不够规定的数。试编程求大、中、小马各有几匹。

解　这是我国古代著名的三元不定方程组问题。它可以有许多组解，即大、中、小马可以有不同的搭配。

法 1　若设大马为 x 匹，中马为 y 匹，小马为 z 匹。则有方程组

$$\begin{cases} x+y+z=100 \\ 3x+2y+0.5z=100 \end{cases}$$

又有

$$\begin{cases} x=-100+1.5z \\ y=200-2.5z \end{cases}$$

由 z 来确定 x、y。当 z 取某整数时，x、y 的正整数解即为所求。

这样，让 z 从 1 变到 100。其实，z 从 50 到 100 就可以了。不必许多空循环。

定义 MWF 为主过程，带一个输入 Z。让 Z 从 50 逐次增 1 取值，确定 X 和 Y 的值。满足条件者打印输出，否则再试。为了取得正整数解，还应调用子过程 ABS。

程序及运行结果如下：

程序 5-2-1

```
TO MWF :Z; 50-100
  IF :Z > 100 STOP
```

```
  MAKE "X ( -100+1.5 * :Z )
  IF NOT :X = ABS INTEGER :X MWF :Z
+ 1 STOP
  MAKE "Y 200- 2.5 * :Z
  IF NOT :Y = ABS INTEGER :Y MWF :Z
+ 1 STOP
  ( PR "X= :X "Y= :Y "Z= :Z )
  MWF :Z + 1
END

?MWF 50
X= 2. Y= 30. Z= 68
X= 5. Y= 25. Z= 70
X= 8. Y= 20. Z= 72
X= 11. Y= 15. Z= 74
X= 14. Y= 10. Z= 76
X= 17. Y= 5. Z= 78
X= 20. Y= 0. Z= 80
```

法 2　建立两个嵌套的递归过程 MWX1 及 MWX2。前者为主过程，带一个输入 X，从 0 开始到 33。后者为子过程，带一个输入 Y，从 0 开始到 50。于是 Z 则取 100-X-Y 进行试算。满足条件时打印输出。

程序及运行结果如下：

程序 5-2-2

```
TO MWX1 :X; 0-33
  IF :X > 33 STOP
```

```
  MWX2 0
  MWX1 :X + 1
END

TO MWX2 :Y; 0-50
  IF :Y > 50 STOP
  MAKE "Z 100 - :X - :Y
  IF 3 * :X + 2 * :Y+ .5* :Z=100
( PR "X= :X "Y= :Y "Z= :Z )
  MWX2 :Y + 1
END

?MWX1 1
X= 2 Y= 30 Z= 68
X= 5 Y= 25 Z= 70
X= 8 Y= 20 Z= 72
X= 11 Y= 15 Z= 74
X= 14 Y= 10 Z= 76
X= 17 Y= 5 Z= 78
X= 20 Y= 0 Z= 80
```

结果是相同的。都可有七种不同的搭配方式。

5-3 角谷猜想

编程验证角谷猜想。对于任意一个数，如果它是偶数，就除以 2；如果是奇数，就乘以 3 并加 1。对结果如此继续，经过有限步后，总能得到 1。

解 程序由两个过程构成，其中 JGCX 为主过程，它调用 YS 子过程。程序只要按题目的要求进行运算就行了。

程序及运行示例如下：

程序 5-3

```
TO JGCX
  MAKE "N FIRST RQ PR "''
  ( PRINT1 "N=  :N "' ' )
  IF :N= 1 PR "'' STOP
  YS :N
END

TO YS :N
  IF :N=  1 PR "'' PR "'' STOP
  IF :N/2 = INTEGER :N / 2 MAKE "
N :N/2( PRINT1 :N "' ' ) YS :N STOP
  MAKE "N :N * 3 +1( PRINT1 :N "' ' )
  YS :N
END

?JGCX
57
N=57 172 86. 43. 130. 65. 196. 98.
49. 148. 74. 37. 112. 56. 28. 14. 7.
22. 11. 34. 17. 52. 26. 13. 40. 20.
10. 5. 16. 8. 4. 2. 1.

?JGCX
```

36

N＝36 18。9。28。14。7。22。11。34。
17。52。26。13。40。20。10。5。16。
8。4。2。1。

5-4 梵塔问题

在贝拿勒斯圣庙里有一块黄铜板，上面插着三根宝石针，其中一根针上穿着64片金片，从上到下一片比一片大，这就是所谓的梵塔。我们将问题一般化，设有三根柱子，柱1上穿有n个大小不同的圆盘。较小的圆盘依次放在较大圆盘的上面。如图5-1所示。欲将圆盘全部搬移到3柱上，大小和排列顺序不变。规则是每次只能移动一个圆盘，而且要始终保证小盘必须在大点圆盘的上面。2柱为辅助柱。试编程序解决这个问题。即圆盘移动的过程。

解 这是一个典型的递归问题，正是LOGO递归过程大显身手的时候。因此，许多LOGO入门及应用的书中，大都以此作例，讲述递归功能。但都复杂烦难，不易理解。这儿，我们介绍只用一行LOGO命令对此问题的解法，可见LOGO递归算法简捷高效的神奇本领。

程序的过程名为FANTA，带4个输入，N表示柱上圆盘的个数。A表示源柱1，B表示目标柱3，C表示辅助柱2。

程序及运行示例如下(结果表明了有3个及4个圆盘的搬移过程)：

程序5-4

```
TO FANTA :N :A :B :C
```

```
  IF :N=0 STOP ELSE FANTA :N-1
:A :C :B ( PR "DISC :N "FROM :A "TO
:B ) FANTA :N-1 :C :B :A
END

?FANTA 3 1 3 2
DISC 1 FROM 1 TO 3
DISC 2 FROM 1 TO 2
DISC 1 FROM 3 TO 2
DISC 3 FROM 1 TO 3
DISC 1 FROM 2 TO 1
DISC 2 FROM 2 TO 3
DISC 1 FROM 1 TO 3

?FANTA 4 1 3 2
DISC 1 FROM 1 TO 2
DISC 2 FROM 1 TO 3
DISC 1 FROM 2 TO 3
DISC 3 FROM 1 TO 2
DISC 1 FROM 3 TO 1
DISC 2 FROM 3 TO 2
DISC 1 FROM 1 TO 2
DISC 4 FROM 1 TO 3
DISC 1 FROM 2 TO 3
DISC 2 FROM 2 TO 1
DISC 1 FROM 3 TO 1
DISC 3 FROM 2 TO 3
```

```
DISC 1 FROM 1 TO 2
DISC 2 FROM 1 TO 3
DISC I FROM 2 TO 3
```

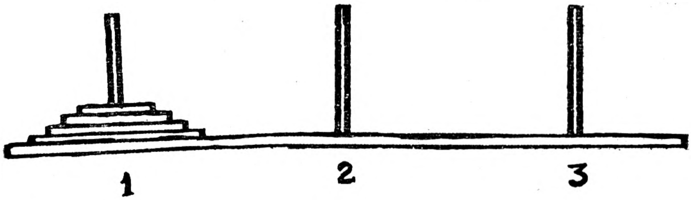


图 5-1

据说对圣庙里的金片，如果每秒钟移动一片，那么，要移完全部 64 个金片，时间会超过地球的年龄。因此，移完梵塔上金片之时，也就是世界的末日，虽属无稽之谈，但足见搬移之难及费时之长。读者可以试试，让机器搬移究竟需要多少时间。

5-5 棋盘奖赏

相传聪明的宰相达依尔，为印度国王舍罕发明了象棋。变化无穷，百玩不厌。国王要奖赏他的宰相。问他需要什么，他回答说："亲爱的国王，什么金银珠宝我都不要，我只要您在棋盘的第一个格子上放一粒小麦，第二个格子上放两粒小麦，第三个格子上放四粒小麦，……。总之，后一个格子上的小麦粒数是前一个格子上的两倍。这样一直放到第六十四个格子。奖给我就行了。"糊涂的国王欣然应允了。

请你想一想，这一共需要多少粒麦子？如果 1 立方米有

1.42×10^8 粒麦子，那么需要多少立方米的麦子。编一个程序求棋盘奖赏的麦粒数及它的体积。

解 要计算 64 格棋盘上所放的麦粒数，实际上就是求 $S=2^0+2^1+2^2+2^3+\cdots+2^{63}$ 之和。

观察和分析这个公式，可以看出有如下的关系：

项数 I	第 I 项的值	前 I 项的和
1	$1=2^0$	1
2	$2^0\times2=2^1$	$1+2^1$
3	$2^1\times2=2^2$	$1+2^1+2^2$
4	$2^2\times2=2^3$	$1+2^1+2^2+2^3$
5	$2^3\times2=2^4$	$1+2^1+2^2+2^3+2^4$
⋮	⋮	⋮
64	$2^{62}\times2=2^{63}$	$1+2^1+2^2+\cdots+2^{63}$

只要用变量 I 记项数，S 记前 I 项之和，它们初值都从 1 开始，用重复命令执行 63 次，给它们赋值。最后 S所存的值即为所求。

过程名 QPJS 1。程序如下：

程序 5-5-1

```
TO QPJS1 :N
  MAKE "S 1 MAKE "I 1
  REPEAT :N [MAKE "I 2*:I MAKE "S :
S+:I]
  PR :S
END
```

还可以使用 GO 命令设置循环模块。用 N 记项数，M 记第 N 项的值，S 记和。过程名为 QPJS2。带两个输入，从

0 和 1 开始。

程序及运行结果如下：

程序 5-5-2

```
TO QPJS2 :N :M; 0 1
  MAKE "S 0
  AA: IF :N>63 (PR "S=  :S) STOP
  MAKE "S :S+:M
  MAKE "N :N+1 MAKE "M :M*2
  GO "AA
END

?QPJS1 63
1.84467E19

?QPJS2 0 1
S=  1.84467E19
T=1.84467E19/1.42E8=1.29906E11
```

由以上的运行结果看，这些麦子的体积约为 1.29906×10^{11} 立方米，相当于在我国 960 万平方公里土地上铺满 1.3 厘米厚的一层。约等于我国几百年的小麦产量。

5-6 约瑟夫斯问题

N 个人围成一圈，从其中某一人开始顺序编号。然后从第一人开始依次报数，凡是报到 M 的人，演一个节目并退出圈子。接着从下一个人又开始报数，报到 M 的人又演节目并退出圈子。如此继续下去，直到 N 个人全都演完节目退

出圈子为止。编程按演出顺序打印出他们在原来圈中的编号。

解 把人数 N 与报数的数 M 作为输入，比较灵活，可以对任意人数及要报的数加以处理。先打印出人数。然后用重复命令 N 次，产生一个表 L，相当于人的编号。再设一记数变量 I，初值为 0。见主过程 JGCH。

子过程 SXDY，给变量 I 每加 1 一次，都使编号表 L 的首项移至表尾。当 I 的值等于 M 时，打印出表的该项，使 I 又从 0 开始，并从表中删去一项。如此循环，直至表空结束。子过程带两个输入，一个是报数M，一个是原编号表。

程序及运行示例（分别是 16 个人，报 5 的退出及 13 人，报 4 的退出）如下：

程序 5-6

```
TO JGCH :N :M
  ( PR "N= :N )
  MAKE "L [ ]
  MAKE "I 0
  REPEAT :N [MAKE "I :I+1 MAKE "L S
E :L :I]
  MAKE "I 0
  SXDY :M :L
END

TO SXDY :M :L
  IF :L = [ ] PR "" TOPLEVEL
  MAKE "I :I + 1
  IF :I = :M ( PRINT1 FIRST :L "' ' )
```

```
MAKE "I 0 SXDY :M BF :L ELSE SX
DY :M ( SE BF :L FIRST :L )
END
?JGCH 16 5
N =  16
5      10      15      4
11      1      8      16
9      3      14      13
2      7      12      6
?JGCH 13 4
N =  13
4      8      12      3
9      1      7      2
11      10      13      6
5
```

5-7 发扑克牌问题

13 张黑桃花色的扑克牌，按顺序 A、2、3、4、5、6、7、8、9、10、J、Q、K 放成一圈，如图 5-2 所示。从 A 开始顺时针(图中箭头所示方向)数牌，数到第 10 张便拿出来。再从下一张牌开始数，数到第 10 张时，又拿出来。这样继续下去直至拿完。编程依次打印出每次拿出的是哪张牌。

解 从 A 开始顺时针方向数牌，数到 10 就拿出来。第一次拿出来的是黑桃 10。又从 J 开始数，第 10 张是黑桃7。拿走以后，下一次又从黑桃 8 开始数，……，最后剩下的是

黑桃 2。

由于 LOGO 表操作的灵活性，先将 13 张牌 A、2、3、4、5、6、7、8、9、10、J、Q、K 作为元素置于表中。并设置一个记数器 I。I 初值从 1 开始，对表中的元素记数。数到 10，就从表中去掉，并打印出来。I 又从 1 开始，继续再数下去。否则，若不是 10，就将数过的牌放在置牌表的最后，并给 I + 1，继续数下去。就这样，一次一次地数，直到表中的元素只剩最后一个，打印出来为止。

此题类似于 5-6 题。不同的是 5-6 题中人的编号是由计算机随机产生的。而此题中表 L 的值是预先给定的。且前者全以数编号，此题中 L 表中还有字作为元素。

图示、程序及运行结果如下。

可以看出，程序主过程 FPK 有一数作输入。表示每次要从第几张牌开始发起，更加灵活和通用。运行结果表示出了分别以 10 及 5 开始的两种情况。

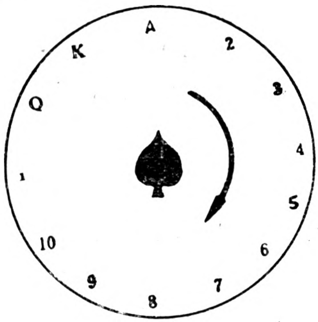


图 5-2

程序 5-7

TO FPK :M

```
  ( PR "M=  :M )
  MAKE "L [A 2 3 4 5 6 7 8 9 10 J Q
  K]
  MAKE "I 0
  PCX :M :L
END

TO PCX :M :L
  IF :L = [ ] PR "'' TOPLEVEL
  MAKE "I :I + 1
  IF :I = :M (PRINT1 FIRST :L "' ' )
MAKE "I 0 PCX :M BF :L ELSE PCX
:M ( SE BF :L FIRST :L )
END

?FPK 10
M= 10
10 7 5 4 6 9 K 8 3 Q A J 2

?FPK 5
M= 5
5 10 2 8 A 9 4 K Q 3 7 J 6
```

5-8 奇妙的素数算式

如图 5-3 所示的一个乘法算式，所有星号位置，皆表示一个素数。编程求出并打印这个算式中所有各位置的数字。

```
    * * *
×)    * *
  * * * *
* * * *
* * * * *
```

图 5-3

解 因为素数数字总共只有 4 个，即 2、3、5、7。在这个算式中，被乘数三位，乘数二位。百位数字和十位数字，都可以取这 4 个素数数字中的任意一个。而两个个位数字，却不能为 2。否则乘积中将有偶数出现。只要被乘数与乘数各位数字确定之后，下边乘积与和也就容易确定了。所以我们可以建立 5 个子过程分别将 4 个素数数字排列组合成被乘数及乘数，被乘数存变量 X 中，乘数存变量 Y 中，乘数个位数与被乘数相乘的结果存变量 Z1 中，乘数十位数与被乘数相乘的结果存变量 Z2 中，最后乘积存变量 Z 中。再由子过程 SSF 对 Z1、Z2、Z 中各个数字分别检验，看是否素数。全部符合条件，则打印输出，否则重新组合再试。程序中对 Z1、Z2 两个中间结果的位数，还随时测试。因而它还调用了子过程 LEN。打印格式由子过程 SS5 附带完成。主过程 SSS，将 4 个素数数字以字形式赋变量 L，并控制调用 SS1。

因为结果是唯一的，所以在打印出结果后即返回命令层。

程序共 8 个过程。程序及运行结果如下：

程序 5-8

```
TO SSS
  MAKE "L "7532
  SS1 :L
```

```
END

TO SS1 :L1
  IF :L1 =" ' STOP
  MAKE "A FIRST :L1
  SS2 :L
  SS1 BF :L1
END

TO SS2 :L2
  IF :L2 =" ' STOP
  MAKE "B FIRST :L2
  SS3 :L
  SS2 BF :L2
END

TO SS3 :L3
  IF :L3 = "'' STOP
  MAKE "C FIRST :L3
  IF :C = 2 SS3 BF :L3
  MAKE "X ( WORD :A :B :C )
  SS4 :L
  SS3 BF :L3
END

TO SS4 :L4
  IF :L4 = "'' STOP
  MAKE "D FIRST :L4
```

```
  MAKE "Z1 :X * :D
  TEST 4 = LEN :Z1
  IFF SS4 BF :L4
  IFT MAKE "T SSF :Z1
  IF :T = "TRUE SS5 :L STOP
  SS4 BF :L4
END

TO SS5 :L5
  IF :L5 = "'' STOP
  MAKE "E FIRST :L5
  IF :E = 2 SS5 BF :L5
  MAKE "Z2 :X * :E
  TEST 4 =LEN :Z2
  IFF SS5 BF :L5
  IFT MAKE "T SSF :Z2
  IF :T = "FALSE SS5 BF :L5 STOP
  MAKE "Y WORD :D :E MAKE "Z :X *
:Y
  MAKE "T SSF :Z
  IF :T = "FALSE SS5 BF :L5 STOP
  ( PR "'    ' :X ) ( PR "'*)  ' :Y )
PR "--------( PR "'  ' :Z1 ) (PR "' '
:Z2 ) PR "--------( PR "' ' :Z ) TOPLEVEL
END
```

```
TO SSF :K
  IF :K = "" OP "TRUE
  MAKE "S FIRST :K
  IF ( ANYOF :S = 2 :S = 3 :S = 5 :S
= 7) OP SSF BF :K
  OP "FALSE
END
```

```
?SSS
     775
*)    33
--------
    2325
   2325
--------
   25575
```

练习：读者还可编程求解一个三位数乘以一位数，积是四位数的乘法算式，所有各个数字全是素数。其结果却不唯一。算式如下图 5-4。

```
      * * *
*)        *
-----------
    * * * *
```

图 5-4

5-9 杨辉三角形

编程打印如图 5-5 所示的杨辉三角。

解 早在 13 世纪，我国南宋时期数学家杨辉，就在他的著作《详解九章算法》（1261 年）中记载了有关二项式系数的研究情况。二项式各系数表示成图 5-5 的形式，即称为

“杨辉三角”。法国数学家帕斯卡也有类似结果（1650 年），所以外国称为“帕斯卡三角”。其实他比杨辉晚了近 400 年。

```
        1
      1   1
    1   2   1
  1   3   3   1
1   4   6   4   1
```

图 5-5

杨辉三角的特点是从第三行开始，每行除了首末两项为 1 之外，其余各项均为上一行两肩位二数之和。根据这一特点，可以利用(FIRST :L)+(FIRST BF :L)产生新行中的数，再加上首末两项的 1，整个新行就产生了。考虑数字在屏幕上的打印位置，可使用光标位置命令，为对三角形行数的限制。在主过程增加一个输入。整个程序含三个过程。主过程 YHSJ按指定位置完成打印各行并调用子过程。子过程 LEN 计表长。子过程 H是程序的关键，以递归方式逐次产生每行各项的值置于变量 LZ中，最后增加首末两项，赋给变量 L，作为新的输入又重新调用 YHSJ 过程，从而打印出新的行。如此十分巧妙地建立了两个相互交叉调用的递归过程。

主过程带两个输入，L 为[1]，H 为屏幕上欲显示的行数。

程序及运行结果如下：

程序 5-9-1

```
TO YHSJ :L :H ; [1] 6
  MAKE "N LEN :L
  IF :N > :H TOPLEVEL
  IF :N > 5 CURSOR 20 - (2 * :N -5 )
:N * 2 -1 PR :L ELSE CURSOR 20
```

```
- :N :N * 2 - 1 PR :L
  MAKE "LZ [ ]
  H :L
END

TO H :L
  IF BF :L = [ ] MAKE " L ( SE FIRST
:L :LZ LAST :L ) YHSJ :L :H
  MAKE "LZ SE :LZ ( FIRST :L ) +
( FIRST BF :L )
  H BF :L
END

?YHSJ [ 1 ] 6
```

```
            1
          1   1
        1   2   1
      1   3   3   1
    1   4   6   4   1
  1   5  10  10   5   1
```

如果不考虑屏幕位置效果，只研究二项式系数的情况，将程序作如下修改，过程名 YHSJ2，只带一个输入，从[1]开始，则可以输出系数的项数更多。

程序及运行示例如下：

程序 5-9-2

```
TO YHSJ2 :L ; [ 1 ]
  AA: PR :L PR "''
  MAKE "LZ [ ]
  BB: IF BF :L = [ ] MAKE "L ( SE FI
```

```
RST :L :LZ LAST :L ) GO "AA
  MAKE "LZ SE :LZ ( FIRST :L ) +
( FIRST BF :L )
  MAKE "L BF :L
  GO "BB
END

?YHSJ2 [ 1 ]
1
1  1
1  2  1
1  3  3  1
1  4  6  4  1
1  5  10  10  5  1
1  6  15  20  15  6  1
1  7  21  35  35  21  7  1
1  8  28  56  70  56  28  8  1
1  9  36  84  126  126  84  36  9  1
```

5-10 孙子定理

试为我国古代著名的孙子定理编一个程序。如有一问题为：一数被 3 除余 2，5 除余 3，7 除余2。求此数最小是几。

此类问题，可以推广到更一般的情况，它可以有任意多个除数和余数的情况。即，有一数 N，被 A 除余 **x**，B 除余 y，C 除余 z，……等等。求此数 N。

解 孙子定理，俗称"鬼谷算"，"隔墙算"，"三齐王暗点兵"等。早在1000多年前，我国古代算书《孙子算经》中，就已有关于这类问题的详细解法。故外国人称为"中国剩余定理"。著名数学家华罗庚先生在《数论导引》中，对这类问题的解法也有详细论述。但用手工计算，工作量一般很大。用计算机来解，变得轻而易举。

先以"韩信点兵"为例，说明几种编程方法，最后给出一种完整通用的程序。

韩信带领不足百人的队伍，若按三人一列排队，最后一列剩一人。若按五人一列排队，最后一列剩二人。若按七人一列排队，则最后一列只有六人。问到底有多少士兵？

法1 利用公式编程。在这儿用

$$n=70\cdot x+21\cdot y+15\cdot z$$

其中 x、y、z 是余数。70为被3除余1的5和7的公倍数；21为被5除余1的3和7的公倍数；15为被7除余1的3和5的公倍数。求出n后，将它再与3、5、7的最小公倍数105比较，若大，逐次减去，得一个符合条件的最小的一个n。（n到底要取哪一个，往往还要由另外的条件确定，因为符合除余的数很多）。如此题中的"不足百人"，便是一个范围。

这样，只要给出余数，就能求出n来。

程序和运行结果如下：

程序 5-10-1

```
TO SUNH :X :Y :Z
  MAKE "N 70 * :X + 21 * :Y + 15 * :Z
  AA: IF :N > 3 * 5 * 7 MAKE "N :N
```

```
- 105 GO "AA ELSE ( PR "N=  :N )
END

?SUNH 1 2 6
N=  97
```

说明：①公式原理具有通用性。②此程序可求用 3、5、7 除的任意余数。③此法人工计算量大，特别是较大数字，而机器做的事情很少。④不通用，换题得重新编程。

法 2 用搜索试除法编程。从第一个余数开始，让所给除数试除，余数相符者，打印输出。只要有一个不符，即加 1 重新开始再试。

程序及运行结果如下。

程序 5-10-2

```
TO SUNS :N ; X
  IF NOT REMAINDER :N 3 = 1 SUNS :N
+ 1 STOP
  IF NOT REMAINDER :N 5 = 2 SUNS :N
+ 1 STOP
  IF NOT REMAINDER :N 7 = 6 SUNS :N
+ 1 STOP
  ( PR "N=  :N )
END

?SUNS 1
N=  97
```

这种方法程序更死。虽由机器多做了工作，但每题必重

新编程，且运行速度较慢。

法 3 通用而且实用程序。原理是先从第一个余数开始，用第二个除数试除，看是否符合第二个余数。不符合，加第一个除数再试，直至符合。然后用第三个除数试除，是否符合第三个余数，不符，加第一、二除数的最小公倍数，再加再试，直至符合为止。还有其它除数和余数，仿此类推。可有任意多个。

此法虽也采用搜索，但不是加 1，而是加前边所有除数的最小公倍数，速度较快。且此法事前并不限定具体的除数和余数的个数，通用性强。

程序由四个过程构成。SUN、PRT、CON 及 GCD。主过程 SUN 赋初值，并调用控制子过程。运行时只要以表按序给出所有的除数和余数。子过程 PRT 打印信息及结果。子过程 CON 反复进行试除和加最小公倍数。子过程 GCD 输出二数最大公约数。

程序及运行示例如下：

程序 5-10-3

```
TO SUN :L
  ND PRT :L
  MAKE "B 1
  MAKE "R FIRST BF :L
  PRINT1 "N(MIN)=
  CON :L
END

TO PRT :L
```

```
  IF :L = [ ] PR "'' STOP
  ( PRINT1 "N/ FIRST :L "····· FIRST
BF :L ) PR "' '
  PRT BF BF :L
END

TO CON :L
  IF :L = [ ] PR :R ( PRINT1 "N=K* :B
"+ :R ) PR "' ' STOP
  AA: IF NOT FIRST BF :L = REMAINDER
:R FIRST :L MAKE "R :R + :B GO "AA
  MAKE "B ( :B * FIRST :L ) / GCD :B
FIRST :L
  CON BF BF :L
END

TO GCD :M :N
  IF REMAINDER :M :N = 0 OP :N
  OP GCD :N REMAINDER :M :N
END
```

例①

```
?SUN [3 2 5 3 7 4]

N/3····· 2
N/5····· 3
N/7····· 4
```

```
N(MIN)=53.
N=K*105.+53.
```

例②

```
?SUN [2 1 3 1 4 1 5 1 6 1 7 0]
N/2…… 1
N/3…… 1
N/4…… 1
N/5…… 1
N/6…… 1
N/7…… 0

N(MIN)=301.
N=K*420.+301.
```

例③

```
?SUN [5 3 7 4 9 2]
N/5…… 3
N/7…… 4
N/9…… 2

N(MIN)=263.
N=K*315.+263.
```

例④

```
?SUN [2 1 3 2 4 1 5 2 6 5 7 3 8 1 9 8]
N/2…… 1
N/3…… 2
```

N/4····· 1

N/5····· 2

N/6····· 5

N/7····· 3

N/8····· 1

N/9····· 8

N(MIN)=17.

N=K*2520.+17.

例⑤

?SUN [5 0 715 10 247 140 391 245 187 109]

N/5····· 0

N/715····· 10

N/247····· 140

N/391····· 245

N/187····· 109

N(MIN)=10020.

N=K*5.31173E6+10020.

可以看出，程序运行后除显示几除余几，一目了然，还显示最小的 N 及所有除数的最小公倍数乘一系数 K，再加上最小的数。K 取正整数。这即便于从许多符合条件的数中根据题目所给的其它条件选择出具体的一个。

运行结果说明：

①仍与韩信点兵同，三列余 2，五列余 3，七列余 4。可求最小的数为 53。其它可为 105 的整数倍加 53。

②所谓的鸡蛋问题。老太婆一篮鸡蛋不知总数，只知两个两个一数，三个三个一数，四个四个一数，五个五个一数，六个六个一数，最后都剩一。只有七个七个数，正好数尽，没有剩余。问老太婆共有多少个鸡蛋？运行结果最少301个。

③5除余3，7除余4，9除余2的最小数是263。5、7、9的最小公倍数是315。

④所谓的爱因斯坦问题。有一楼梯，若一步跨两阶，最后剩一阶；一步跨三阶，最后剩两阶；一步跨四阶，最后剩一阶；一步跨五阶，最后剩两阶；一步跨六阶，最后剩五阶；一步跨七阶，最后剩三阶；一步跨八阶，最后剩一阶；若一步跨九阶，则最后剩八阶。问楼梯到底有几阶。结果最少17阶。当然，人一步是跨不了九阶的。五阶恐怕也有困难。

⑤最后一个运行结果，是《数论导引》中孙子定理部分的一个习题。大意是一数5可整除，715除余10，247除余140，391除余245，187除余109。求此数。该数最小为10020。而5个除数的最小公倍数为5311730。若用笔算，还真得费点时间和力气。

由这一问题的结果看，还说明这样两个问题：一，输入次序与除数大小顺序无关；二，除数是否互质，并不限制。如247比715小，却在后；而715与5并不互质。有些地方对此加以限制，是不必要的。

六 高精度计算

6-1 高精度加法

LOGO 高精度加法程序。

解 LOGO数值计算中，整数，最大输出10位，还不能大于 $2^{31}-1$，即2147483647。不能小于 -2^{31}，即 -2147483648。对实数，绝对值在 10^{38} 与 10^{-38} 之间，只有6位有效数字。用指数记数法输出。对于更多位数的运算，便得不到精确值。为了实现高精度的精确计算，我们编写如下程序，进行高精度的加法运算。

程序由三个过程构成。AD1、AD2 及LEN。其中AD1为主过程。它带两个输入，即要加的二数。但要以字的形式给出。过程先对两个加数计长，并把较长者作为被加数，较短者作为加数，调用子过程 AD2 逐位相加。最后打印出和式与结果。子过程 AD2 是主要过程，完成逐位相加并记和。它带两个输入，即前两个加数。较长者在前。逐位相加之和送 C，进位送 D，S 存放临时结果。最后将整个结果送 T，打印输出。程序还调用字函数过程 LEN 来计字长。

整个程序及运行示例如下：

程序 6-1

```
TO AD1 :M :N
  MAKE "S "' ' MAKE "D 0
```

```
  TEST LEN :M > LEN :N
  IFT MAKE "T AD2 :M :N
  IFF MAKE "T AD2 :N :M
  IF FIRST :T= 0 MAKE "T BF :T
  PR ( WORD :M "+ :N "= :T )
END

TO AD2 :A :B
  IF :B="' ' MAKE "B 0
  IF :A ="' ' OP WORD :D :S
  MAKE "C ( ( LAST :A ) + ( LAST :B)
+ :D )
  MAKE "D QUOTIENT :C 10
  MAKE "S WORD LAST :C :S
  OP AD2 BL :A BL :B
END

?AD1 "12345678901 "9876543210
12345678901+9876543210=22222222111

?AD1 "123456789 "864197532
123456789+864197532=987654321
```

6-2 高精度减法

LOGO 高精度减法程序。

解 建立三个过程，ST1、ST2、ST3。其中 ST1 为主过程。它的两个输入为被减数和减数。ST2 的两个输入与

它相同。过程中几个变量的意义如下：H为被减数最末一位数码，K为减数末位数码。D为借位数，S放相减之差的临时结果。C为逐位的差。最后结果送T，并由ST3整理检查，去掉有效数字前面的0，打印输出。

程序及运行示例如下：

程序 6-2

```
TO ST1 :M :N
  MAKE "D 0 MAKE "S "''
  MAKE "T ( ST2 :M :N )
  MAKE "R ( ST3 :T )
  PR ( WORD :M "- :N "= :R )
END

TO ST2 :A :B
  IF :B = "'' MAKE "B 0
  IF :A = "'' OP :S
  MAKE "H LAST :A
  MAKE "K ( ( LAST :B ) + :D )
  IF :H < :K MAKE "D 1 ELSE MAKE "D
0
  MAKE "C ( WORD :D LAST :H ) - :K
  MAKE "S WORD ( LAST :C ) :S
  OP ST2 BL :A BL :B
END

TO ST3 :T
  IF :T = "'' OP 0
```

```
  IF NOT FIRST :T = 0 OP :T
  OP ST3 BF :T
END

?ST1 "98765432101 "12345678901
98765432101－12345678901＝86419753200
```

6-3　高精度乘法

LOGO 高精度乘法程序。

解　要求两个多位数乘积，并且输出乘积的各位数的精确值，直接利用 LOGO 数字相乘是不行的。我们采用同笔算竖式相同的设计思想。把需要相乘的两个多位数，都以字的形式输入。然后从右向左，逐位取出单独相乘，把相乘的结果，再分离出个位数和十位数。十位数作为进位，与下次相乘的结果相加。个位数再以字存入结果变量，待后输出。这样，用乘数的一位，遍乘被乘数各位后有一个中间结果，错位加入最后的结果中，就能得到所有各位的精确值。

关于多位数的乘法，调用高精度加法程序，并作少许修改，作为一个子过程使用。而程序的技巧在于，增设一个变量，巧妙地解决每次相乘之后中间结果的错位相加。具体见主过程 GCK第六行，子过程 GCF 第四行及第七行。

程序由六个过程构成。子过程 AD1、LEN 及 AD2，用来完成两个多位数的求和（见高精度加法程序）。子过程 CHE 完成一位数乘多位数。子过程GCF，从后向前依次逐位递归取出乘数各位（最末一位除外）。主过程是 GCK。它给变量赋初值，并调用 CHE 子过程一次，求出乘数末一位与被乘

数的乘积，然后打印出算式及整个程序运行的最后结果。主过程带两个输入，即以字形式给出的被乘数和乘数。

程序运行，最多可求乘积达48位。

程序及示例如下：

程序6-3

```
TO GCK :K :L
  ( PRINT1 :K "* :L "= )
  MAKE "P 0 MAKE "Q "' '
  MAKE "X CHE :K LAST :L
  IF FIRST :X = 0 MAKE "X BF :X
  MAKE "O 0
  PR GCF :K BL :L
END

TO CHE :U :V
  IF :U = "' ' OP WORD :P :Q
  MAKE "W :P + :V * LAST :U
  MAKE "P QUOTIENT :W 10
  MAKE "Q WORD LAST :W :Q
  OP CHE BL :U :V
END

TO GCF :I :J
  IF :J = "' ' OP :X
  MAKE "P 0 MAKE "Q "''
  MAKE "O :O + 1
  MAKE "Y CHE :I LAST :J
```

```
  IF FIRST :Y = 0 MAKE "Y BF :Y
  REPEAT :O [MAKE "Y WORD :Y "0]
  MAKE "X AD1 :X :Y
  OP GCF :I BL :J
END

TO AD1 :M :N
  MAKE "S "" MAKE "D 0
  TEST LEN :M > LEN :N
  IFT MAKE "T AD2 :M :N
  IFF MAKE "T AD2 :N :M
  IF FIRST :T = 0 MAKE "T BF :T
  OP :T
END

TO LEN :K
  MAKE "H 0
  AA: IF :K = "" OP :H
  MAKE "H :H + 1
  MAKE "K BF :K GO "AA
END

TO AD2 :A :B
  IF :B = "" MAKE "B 0
  IF :A = "" OP WORD :D :S
  MAKE "C ( ( LAST :A ) + (LAST :B )
+ :D )
```

```
  MAKE "D QUOTIENT :C 10
  MAKE "S WORD LAST :C :S
  OP AD2 BL :A BL :B
END
```

```
?GCK "12345678909876543 21 "123456789
12345678909876543 21*123456789=1524
1578762383784111 2635269
```

6-4　高精度阶乘

LOGO 高精度阶乘程序。

解　高精度阶乘的设计思想，与高精度乘法相同。因为阶乘是求自然数的连乘积，这样只要能多次使用两个多位数相乘程序就行了。

于是，在高精度乘法程序的基础上，作少许修改。具体作法是删去乘法程序主过程 GCK 的第五行，并将第七行中的打印命令改为输出命令。这样，将它作为一个子过程。另外重新定义一个主过程，以控制整个程序的运行。如主过程为 GJC。它带一个输入，就是要求的阶乘数。

乘法程序的其它五个子过程，仍然作为子过程调用而不作改动。这样，阶乘程序共有七个过程。过程名单如下：

```
?POTS
TO GJC :N
TO GCK :K :L
TO CHE :U :V
TO GCF :I :J
```

```
TO AD1 :M :N
TO LEN :K
TO AD2 :A :B
```

主过程 GJC 及修改后的子过程 GCK 清单及运行示例如下。

在 CEC-I 中华学习机上，运行此程序最大可求出 45!，输出精确结果 57 位数。

程序 6-4

```
TO GJC :N
  MAKE "JC "1 MAKE "I 2
  AA: IF :I > :N (PR :N "!= :JC ) STOP
  MAKE "JC GCK :JC :I
  MAKE "I :I+ 1 GO "AA
END

TO GCK :K :L
  MAKE "P 0
  MAKE "Q "' '
  MAKE "X CHE :K LAST :L
  IF FIRST :X = 0 MAKE "X BF :X
  MAKE "O 0
  OP GCF :K BL :L
END

?GJC 10
10 !=3628800

?GJC 33
```

33 ! = 8683317618811886495518194401280000000

6-5 高精度乘方

LOGO 高精度乘方运算程序。

解 乘方运算，对于整数的正整数次幂，定义为相同因数自乘的次数。它还是连乘问题，于是仍然需要进行乘法运算及错位相加的加法运算。如同阶乘一样，还需保留六个子过程，而重新定义一个能输入运算所需的底数和指数，并对其它子过程进行控制的主过程。

建立主过程 GPW。它也类似于数值函数过程 POW。区别在于，它是主过程，不再输出数值。另外，输入底数 B 是以字形式给出，所以此程序只对底数是正的多位数，而指数只能是正整数和零适用。

下面是主过程 GPW。其它子过程同高精度阶乘，不再列出。运行示例与高精度乘法比较，结果完全一样(见GPW ″12345678 2与 PR GCK ″12345678 ″12345678)。

程序及运行示例如下：

程序 6-5

```
TO GPW :B :E
  IF :E = 0 OP 1
  IF :E = 1 OP :B
  MAKE "PW :B
  MAKE "I 2
  AA: IF :I > :E ( PR :B CHAR 94 :E
"= :PW ) STOP
```

```
  MAKE "PW GCK :PW :B
  MAKE "I :I + 1 GO "AA
END
```

```
?GPW "12345678 2
12345678∧ 2 = 152415765279684

?GPW "12345678 3
12345678 ∧ 3 = 1881675960266558605752

?PR GCK "12345678 "12345678
152415765279684
```

程序共由七个过程组成，过程名单如下：

```
?POTS
TO GPW :B :E
TO GCK :K :L
TO CHE :U :V
TO GCF :I :J
TO AD1 :M :N
TO LEN :K
TO AD2 :A :B
```

6-6 高精度除法

LOGO 高精度除法程序。

解 除法运算中，LOGO 最多输出六位有效数字。如355/113，是大家熟悉的一个分数。它是我国古代伟大数学家祖冲之发现的圆周率“密率”。它的前九位是3.14159292…，

以后是多少呢？由于输出形式限制，无法得到。下面编一个程序，执行时只要输入被除数、除数，及商数所需要小数后的位数，计算机就会按要求的精度输出小数。

程序由如下的四个过程组成。DV1、DV2、DV3和DV4。其中DV1为主过程。它调用子过程DV2和DV3。逐位相除。DV3又调用DV4，处理小数后各位。

程序及运行示例如下。可以看出，其结果求出了355/113的小数后20位。

程序 6-6

```
TO DV1 :A1 :A2 :A3
  IF :A2 = 0 PR "E STOP
  ( PRINT1 :A1 "/ :A2 "= )
  IF :A1 < :A2 PRINT1 [0.] DV3 :A1
:A2 :A3 1 1
  DV2 :A1 :A2 :A3 1
END

TO DV2 :A1 :A2 :A3 :A4
  MAKE "A5 :A1 - :A2 * :A4
  IF :A5 = 0 PRINT1 :A4 STOP
  TEST :A5 < :A2
  IFT PRINT1 :A4 PRINT1 [ . ] DV3 :A5
:A2 :A3 1 1
  IFF DV2 :A1 :A2 :A3 :A4 + 1
END
```

```
TO DV3 :A1 :A2 :A3 :A4 :A5
  IF :A4 > :A3 + 1 PR [ ] TOPLEVEL
  TEST 10 * :A1 < :A2
  IFT PRINT1 0 MAKE "A1 :A1 * 10
MAKE "A4 :A4 + 1
  IFF MAKE "A6 10 * :A1 - :A2 * :A5
  DV4
  DV3 :A1 :A2 :A3 :A4 :A5
END

TO DV4
  IF :A6 = 0 ( PRINT1 :A5 0 ) PR [ ]
  TOPLEVEL
  TEST :A6 < :A2
  IFT PRINT1 :A5 MAKE "A1 :A6 MAKE
"A4 :A4 + 1 MAKE "A5 1
  IFF MAKE "A5 :A5 + 1
END

?DV1 "9876543210 "9999999999 10
9876543210/9999999999=0.98765444439

?DV1 355 113 20
355/113=3.141592920353982300884
```

七　数制转换

7-1　10进制数转换成2、8、16进制数

试编一程序，可将任一10进制整数转换成2、8或16进制的数。

解　将10进制整数转换成其它任一进制的数，只要采用“除R取余法”。R为要转换成的进制基数。对10以内的进制，用0～9个数码即可。对大于10进制的其它进制，必须有A、B、C、D、E、F等字母表示的数，要考虑到字符转换，或者用ASCII码值表示这些字符。

程序由四个过程构成。A为主过程，B、C、D为子过程。

程序如下：

程序 7-1

```
TO A :N :B ; 10--->2,8,16
  PR ″''
  ( PRINT1 :N ″(10)---> )
  (PRINT1 C :N :B ″( :B ″) )
  PR ″'' PR ″'' PR ″''
END

TO B :N :B
  IF :N = 0 OP 0
```

```
  OP WORD B ( QUOTIENT :N :B ) :B D
  REMAINDER :N :B
END

TO C :N :B
  IF :B > 16 PR "ERR STOP
  IF :N = 0 OP 0
  OP BF B :N :B
END

TO D :N
  IF :N < 10 OP :N
  IF :N = 10 OP "A
  IF :N = 11 OP "B
  IF :N = 12  OP "C
  IF :N = 13 OP "D
  IF :N = 14 OP "E
  IF :N = 15 OP "F
END
```

7-2　0～255 间 10 进制数对应转换成2、8、16进制数

打印出 0～255 间 10 进制整数转换成 2、8、16 进制数的对照表。

解　数进制转换，方法较多。此题指定范围，且要连续打印出对照表。我们只需定义一个主过程 A，让它完成对照

打印并根据范围，确定终止条件。其它子过程 B、C、D则与 7-1 题完全相同。而且，主过程只需两个输入，一个是10进制的数，一个是要转换的新进制。

程序及运行示例的部分结果如下：

程序 7-2

```
TO A :N :B ; 0-255--->2,8,16
  IF :N > 255 STOP
  ( PRINT1 :N "(10)---> )
  ( PRINT1 C :N :B "( :B ") )
  PR "' ' PR "' '
  A :N + 1 :B
END

?A 0 2
0(10)--->0(2)
1(10)--->1(2)
2(10)--->10(2)
3(10)--->11(2)
4(10)--->100(2)
5(10)--->101(2)
6(10)--->110(2)
7(10)--->111(2)
8(10)--->1000(2)
9(10)--->1001(2)
10(10)--->1010(2)

?A 0 8
```

```
0(10)--->0(8)
1(10)--->1(8)
2(10)--->2(8)
3(10)--->3(8)
4(10)--->4(8)
5(10)--->5(8)
6(10)--->6(8)
7(10)--->
STOPPED!, IN LINE
   OP WORD B (QUOTIENT :N :B ) :B D
      REMAINDER :N :B
 AT LEVEL 11 OF B.

?A 0 16
0(10)--->0(16)
1(10)--->1(16)
2(10)--->2(16)
3(10)--->3(16)
4(10)--->4(16)
5(10)--->5(16)
6(10)--->6(16)
7(10)--->7(16)
8(10)--->8(16)
9(10)--->9(16)
10(10)--->A(16)
11(10)--->
```

```
STOPPED!, IN LINE
 ( PRINT1 C :N :B "( :B ") )
AT LEVEL 13 OF A.
```

7-3 任意两种进制间数的转换

编写一个程序，实现2、8、10、16几种进制中的任意两种进制间整数的相互转换。

解 数进制间的转换，可以按照按权展开写出它们的等价式。无论哪种进制的数，按权展开运算之后，都得到它的10进制表示。而10进制整数转换成其它进制，采用“除N取余法”很容易办到。所以，我们采用无论哪种数，先把它转换成10进制数，然后再由10进制数，转换新的进制数。

下面定义六个过程。A、B、C及LEN、MID＄和PICK。

其中A为主过程，它带三个输入。第一个是要转换的数，第二个是原进制，第三个是新进制。主过程中的变量S存第一个输入的10进制表示。变量E＄表示转换成的新进制的数，以LOGO字表示。它调用B、C之后，打印出运行结果。

B过程完成把输入的数变成10进制，不管原来是什么数。过程C完成把10进制数换成新进制。LEN、MID＄和PICK是字函数过程。

要说明的是：一，2、8、10进制的数，都只用0～9间的数字符号。把它们当作字符的时候，用该字符的ASCII码值减48就行了。如"9（即字9）。ASCII "9＝57，而数9＝（ASCII "9）－48。二，16进制的数除0～9之外，还

要用 A、B、C、D、E、F 表示 10～15。转换中要用英文字母的 ASCII 码值减 55。如 10 = (ASCII ″A) − 55 = 65 − 55。

下面是 A、B、C 三个过程的清单及运行示例。实际使用该程序时还应有 LEN、MID$ 与PICK 三个过程。

程序 7-3

```
TO A :A$ :B :C
  MAKE "N LEN :A$
  MAKE "K 1
  MAKE "S 0
  MAKE "E$ "' '
  B :N
  C
  (PR :A$ "( :B ")= :E$ "( :C ") )
END

TO B :I
  IF :I = 0 STOP
  MAKE "D ASCII MID$ :A$ :I 1
  MAKE "X 55
  IF :D < 58 MAKE "X 48
  MAKE "D :D - :X
  MAKE "S :S + :K * :D
  MAKE "K :K * :B
  B :I - 1
END

TO C
```

```
    MAKE "D REMAINDER :S :C
    MAKE "X 48
    IF :D > 9 MAKE "X 55
    MAKE "D :D + :X
    MAKE "E$ WORD CHAR :D :E$
    MAKE "S QUOTIENT :S :C
    IF :S > 0 C STOP
END
```

```
?A 145 10 2
145 ( 10 )= 10010001 ( 2 )

?A 145 10 8
145 ( 10 )= 221 ( 8 )
```

该程序由于采用 ASCII 函数命令，不仅使程序简短，而且，使用该程序可以实现对小于 16 进制的任两种进制间的数字转换，不仅仅限于 2、8、10 和 16 四种。可见如下的运行结果：

```
?A 145 10 16
145 ( 10 )= 91 ( 16 )

?A "EFFF 16 10
EFFF ( 16 )= 61439 ( 10 )

?A 456 8 10
456 (8)= 302 ( 10 )

?A 1100101 2 10
```

1100101 (2) = 101 (10)

?A 279 10 5

279 (10) = 2104 (5)

?A 333 9 10

333 (9) = 273 (10)

?A 333 9 16

333 (9) = 111 (16)

八 比较与排序

8-1 打印最大的一个数

输入一组数，要求打印出其中最大（或最小）的一个。

解法1 主过程 ZDS，以表为输入。表中元素，即已知的一组数据。它的功能是将输入表中所有各元素取出由子过程比较一遍。子过程MAX，比较二数，输出较大的一个。

程序及其运行示例如下：

程序 8-1-1

```
TO ZDS :L
  IF BF :L = [ ] OP FIRST :L
  OP MAX FIRST :L ZDS BF :L
END

TO MAX :X :Y
  IF :X > :Y OP :X
  OP :Y
END

?PR ZDS [65 43 89 21 77 35]
89
```

解法2 建立三个过程，主过程 AD，两个输入，数据个数及数据表。变量 M 取表中第一个元素。变量 I 记位

置。子过程 BD 将表中元素遍比之后存入变量 M 中，它还调用 PICK 过程。这个程序可以同时输出最大数在原表中的位置。

程序及运行示例如下：

程序 8-1-2

```
TO AD :N :L
  IF :L = [ ] STOP
  MAKE "M FIRST :L
  MAKE "I 1
  BD 1 :L
END

TO BD :K :L
  IF :K > :N ( PRINT1 :M "'-----' )
  PR :I STOP
  IF :M < PICK :K :L MAKE "M PICK
:K :L MAKE "I :K
  BD :K + 1 :L
END

?AD 5 [77 55 44 11 99]
99-----5
```

若要打印输出最小的一个数，只要在条件关系式处作相反处理。

8-2　同时打印最大和最小的一个数

输入一组数，同时打印出其中最大的一个及最小的一个

和它们各自原来的序号。

解 在比较之前，先设置两个变量，存放最大数和最小数。均可将表中第一个数值作为初值。比较后，根据情况决定是否交换。再用两个变量来记最大数和最小数在表中的位置。

定义两个过程 A 和 B。主过程 A，用变量 D 记最大数，I 记最大数序号；变量 X 记最小数，J 记最小数序号。I、J 的初值均为 1，D、X 的初值都是表中第一个元素。

子过程 B 完成比较和打印，并调用子过程 PICK。程序及运行示例如下：

程序 8-2

```
TO A :N :L
  IF :L = [ ] STOP
  MAKE "D FIRST :L
  MAKE "I 1
  MAKE "X FIRST :L
  MAKE "J 1
  B 1 :L
END

TO B :K :L
  IF :K > :N ( PRINT1 "'ZUIDA ' :D "'
  -----' ) PR :I (PRINT1 "'ZUIXAO'
  :X "'-----' ) PR :J STOP
  MAKE "R PICK :K :L
  IF :D < :R MAKE "D :R MAKE "I :K
```

```
  IF :X > :R MAKE "X :R MAKE "J :K
  B :K + 1 :L
END
?A 5 [12 11 78 63 57]
ZUIDA   78-----3
ZUIXAO 11-----2
```

8-3　数 字 排 序

对已给出的一组数据按从大到小(或从小到大)排序。

解　排序问题，是比较典型的数据处理问题。在日常生活中经常遇到此类问题。如学生名次、运动员名次，都要根据成绩及记录排出。

排序方法的原理很多。经常使用的有选择法、冒泡法及最小元素法等。我们在此使用选择法。它是先把第一个元素取出，如存入变量M，然后让其逐个与其它元素相比，不大于它的，不动。大于它的，则互相交换。最后，M中选择出最大的一个。其后，又遍比一遍之后，再选出次大的一个等等。

为此，我们编制四个过程 A、B、E、P。A 为主过程，其余三个为子过程。其中子过程 P 按指定的 N 输出表 L 的项。子过程 E 按指定的 N 删去表 L 的项。子过程 B 是关键，它完成M中的数与表 L 中所有元素遍比一次的任务，并把其中最大的一个换存入M中。

主过程 A 给变量M及项数 I 赋初值。它带两个输入，一个是要排序的数据，一个是数据个数。它调用子过程 B，及

经 E 删除一个最大元素后对新表的递归调用。

程序及运行示例如下：

程序 8-3

```
TO A :N :L
  IF :L = [ ] PR "" STOP
  MAKE "M FIRST :L
  MAKE "I 1
  B 1 :L
  A :N - 1 E :I :L
END

TO B :K :L
  IF :K > :N ( PRINT1 :M "' ' ) STOP
  MAKE "R P :K :L
  TEST NOT :M < :R
  IFT B :K + 1 :L
  IFF MAKE "M :R MAKE "I :K B :K +
1 :L
END

TO E :N :L
  IF :L = [ ] STOP
  IF :N = 1 OP BF :L
  OP SE FIRST :L E :N - 1 BF :L
END

TO P :N :L
  IF :N = 1 OP FIRST :L
```

```
  OP P :N - 1 BF :L
END

?A 10 [76 84 99 90 85 79 82 88 69 70]
99 90 88 85 84 82 79 76 70 69
```

从小到大排序，只要将遍比过程中的判断条件改为相反即可。

让机器随机产生一组数据排序，只要增加一产生随机数过程就行了。作为练习，请读者修改程序，对随机数排序。

8-4 字母按ASCII码排序

随机产生一组英文字母，按 ASCII 码排序。

解 这个问题，首先要解决用随机函数产生英文字母。用 RANDOM 26，可以产生 0～25 的整数，该数加 65，即在 65～90 间，包括 65 与 90。然后将它们作为 CHAR 命令的输入，便可输出 26 个英文字母。

其次，表中元素为英文字母，在比较时不能直接使用，再用 ASCII 命令反转成数，然后再行比较，问题便完满解决了。

程序由 A、C、B、E、P 五个过程构成。原理与功能类似于 8-3 题，只要经过两次转换就行了。

程序及运行示例如下：

程序 8-4

```
TO A :N
  MAKE "L [ ]
```

```
  REPEAT :N [MAKE "L (SE :L (CHAR
65+RANDOM 26) ) ]
  PR :L
  C :N :L
END

TO B :K :L
  IF :K > :N ( PRINT1 CHAR :M "' ')
STOP
  TEST NOT :M > ASCII P :K :L
  IFT B :K + 1 :L
  IFF MAKE "M ASCII P :K :L MAKE "I
:K B :K + 1 :L
END

TO C :N :L
  IF :L = [ ] PR "'' STOP
  MAKE "M ASCII FIRST :L
  MAKE "I 1
  B 1 :L
  C :N - 1 E :I :L
END

TO E :N :L
  IF :L = [ ] STOP
  IF :N = 1 OP BF :L
  OP (SE FIRST :L E :N - 1 BF :L )
```

```
END

TO P :N :L
  IF :N = 1 OP FIRST :L
  OP P :N - 1 BF :L
END

?A 10
Y R A B V G I Z L B
A B B G I L R V Y Z
```

读者可练习编写直接输入一组英文字母，让计算机按ASCII码进行排序。

8-5　随机字母组的字典排序

随机产生给定个数的多项多字母字的字典排序。

解　此题虽较复杂，但道理与前几题类似。不过多几道手续罢了。在前几题的基础上，定义如下八个过程：A、B C、D、E、I、O、P。它们的功能是：

A 为主过程，给表 L 赋初值，并调用子过程 D、C。它有一个输入，指定表的长。

D 过程随机产生表 L 的各项内容。每项由 3～5 个字符组成。

C 过程给变量M1、M、I 赋初值，分别存表的第一项、第一项的第一个字符，及项数 1。然后控制比较过程 B，及删项过程 E。

B 过程用以比较。它调用过程 P，P 输出表中指定项。

P 又调用 O 过程。O 输出一项中的指定字符。B 也调用 I过程，给变量 M1 赋值。I 过程输出指定的表中的项。在所判断的条件不真的情况下改变变量 M1、M 及 I 的值，并增值递归调用本身。

程序及运行示例如下：

程序 8-5

```
TO A :N
  MAKE "L [ ]
  D 1 :N
  C :N :L
  PR "'
END

TO B :K :L
  IF :K > :N ( PRINT1 :M1 "' ' ) STOP
  TEST NOT :M > ASCII P :K :L
  IFT B :K + 1 :L
  IFF MAKE "M1 I :K :L MAKE "M ASCII
P :K :L MAKE "I :K B :K +1 :L
END

TO C :N :L
  IF :L = [ ] STOP
  MAKE "M ASCII FIRST FIRST :L
  MAKE "M1 FIRST :L
  MAKE "I 1
  B 1 :L
```

```
  C :N -1 E :I :L
END

TO D :K :N
  IF :K > :N PR :L STOP
  MAKE "I 3 + RANDOM 3
  MAKE "M "' '
  REPEAT :I [MAKE "M WORD :M CHAR
65+RANDOM 26]
  MAKE "L SE :L :M
  D :K + 1 :N
END

TO E :N :L
  IF :L = [ ] STOP
  IF :N = 1 OP BF :L
  OP ( SE FIRST :L E :N -1 BF :L )
END

TO I :N :L
  IF :N = 1 OP FIRST :L
  OP I :N - 1 BF :L
END

TO O :N :L
  IF :N = 1 OP FIRST :L
  OP P :N -1 BF :L
```

```
END

TO P :N :L
  IF :N = 1 OP O 1 FIRST :L
  OP P :N - 1 BF :L
END

?A 5
QDPFF GUT SGJVZ CWE ZJHHC
CWE GUT QDPFF SGJVZ ZJHHC
```

8-6 英文人名字典排序

对给定的英文或汉语拼音人名（国名、地名、杂志报刊名）进行字典排序。

解 对给定的英文字母人名、国名、地名或杂志报刊名（汉语拼音字母同）进行字典排序。其方法与8-5题类似。只要删去随机产生字符项的过程 D，并将主过程 A 稍加修改，使它带两个输入，其一为输入表的项数，另一为要进行排序的内容，以表给出。其余子过程照旧不变。

为清楚起见，把程序的全部过程 A、C、B、E、I、O、P 全列于下。并附运行示例。

程序 8-6

```
TO A :N :L
  PR :L
  C :N :L
  PR "' '
```

```
END

TO C :N :L
  IF :L = [ ] STOP
  MAKE "M ASCII FIRST FIRST :L
  MAKE "M1 FIRST :L
  MAKE "I 1
  B 1 :L
  C :N - 1 E :I :L
END

TO B :K :L
  IF :K > :N ( PRINT1 :M1 "' ' ) STOP
  TEST NOT :M > ASCII P :K :L
  IFT B :K + 1 :L
  IFF MAKE "M1 I :K :L MAKE "M ASCII
P :K :L MAKE "I :K B :K + 1 :L
END

TO E :N :L
  IF :L = [ ] STOP
  IF :N = 1 OP BF :L
  OP ( SE FIRST :L E :N - 1 BF :L )
END

TO I :N :L
  IF :N = 1 OP FIRST :L
  OP I :N - 1 BF :L
```

```
END

TO O :N :L
  IF :N = 1 OP FIRST :L
  OP P :N - 1 BF :L
END

TO P :N :L
  IF :N = 1 OP O 1 FIRST :L
  OP P :N - 1 BF :L
END

?A 5 [KMN TREW FDS ABC UVW]
KMN TREW FDS ABC UVW
ABC FDS KMN TREW UVW
```

8-7 映射排序应用

映射排序法在 LOGO 排序中的应用。

解 像 8-3 题与 8-4 题，无论以键盘输入数据，还是让计算机随机产生原始数据，然后排序，其编程方法是对的，但不太实用。据作者试验，用 8-3 题(主过程 A)，随机产生 40 个数，排序完毕共需 5 分钟。若随机产生 100 个数，排序便不能进行了。为此，我们改进程序，应用映射法排序。它可以大大增加数据的个数，而接近实际使用。

原理是这样的：首先设置一个存放原始数据的数表 LN，其长为 N，即 N 个数据元素。再设置一个存放相 同 数 数据

个数的数表 TH，其长为 H，H 为估计的最大数据(如随机产生 0～100 间的整数，则 H＝100)。开始，TH 各元素均为 0。在随机产生数据给 LN赋值的同时，在表 TH 中就“映射”出相同数据的个数。实际就是在 LN 中产生一个随机数，在 TH 中便按“位置”加 1。如下图。

表 LN [… | 98 | … | 98 | … … | 55 | 55 | … | 55 | …]

第10项　第21项　第64项第65项　第80项

表 TH [… … … | 3 | … … … … … | 2 | …]

第55项　　　　第98项

可以看出，LN(10)＝98,即表 LN 第 10 项元素是 98，同理 LN(21)＝98。共有两个 98。于是在表 TH 第 98 项中元素为 2。即 TH(98)＝2。这样 TH(55)＝3，就说 LN 中有 3 个相同的 55。如果 TH 中某项为 0，则说明 LN 中无此元素。上图中若 TH(100)＝0，TH(99)＝0，则第 98 项的两个 98 都可认定为最大的。

最后，设置一个数表 DN，存放排好序的元素并打印输出。以上，可见程序中的主过程 A。另外两个子过程 P 和 W，则分别用来输出表中指定项，及写入有变化的项。

运行此程序，对随机产生的 300 个 50～100 的整数，大约需要 30 多分钟。

程序及运行示例如下：

程序 8-7

```
TO A :N :H
  MAKE "LN [ ] MAKE "TH [ ]
  REPEAT :H [MAKE "TH SE :TH 0]
  REPEAT :N [MAKE "LN SE :LN  50+RAN
```

```
DOM 51 MAKE "R 1+(P LAST :LN :TH)
MAKE "TH W LAST :LN :TH]
  PR :LN MAKE "DN [ ]
  AA: IF :TH = [ ] PR :DN STOP
  MAKE "D LAST :TH
  IF :D = 0 MAKE "TH BL :TH MAKE "H
:H - 1 GO "AA
  REPEAT :D [MAKE "DN SE :DN :H]
  MAKE "TH BL :TH MAKE "H :H - 1 GO
"AA
END

TO W :N :L
  MAKE "LQ [ ]
  REPEAT :N - 1 [MAKE "LQ SE :LQ
FIRST :L MAKE "L BF :L]
  OP ( SE :LQ :R BF :L )
END

TO P :N :L
  REPEAT :N - 1 [MAKE "L BF :L]
  OP FIRST :L
END

?A 15 100
77 60 93 75 69 63 60 96 77 98 83 50 52 92 62
98 96 93 92 83 77 77 75 69 63 62 60 60 52 50
```

九 统计应用

9-1 选票统计①

有王五、张三、李四三人为候选人。在三人当中选一名班长，得票最多者当选。编程统计各人所得票数。

解 程序先打印出候选人数及姓名。设变量 W、Z、L 存放各人所得票数。先清零。然后由键盘输入一张一张的选票(字 W、Z、L)，赋值给变量 P，然后检查 P 的值，分别给三个变量加 1。待全部选票输入完毕，按 AAA 结束打印各人票数。程序中用“CF:”作为小循环模块，逐一统计选票。

程序及运行示例如下：

程序 9-1

```
TO TJXP
  ( PR "N=3 "------ "W "Z "L )
  MAKE "W 0
  MAKE "Z 0
  MAKE "L 0
  CF: MAKE "P FIRST RQ
  IF :P = "AAA ( PR "W= :W "Z= :Z "L
= :L ) STOP
  IF :P = "W MAKE "W :W + 1
  IF :P = "Z MAKE "Z :Z + 1
```

```
  IF :P = "L MAKE "L :L + 1
  GO "CF
END

?TJXP
N=3 ----- W Z L
W= 8  Z= 6  L= 8
```

9-2 选票统计②

某单位进行选举，有8位候选人，他们的代号分别为A、B、C、D、E、F、G、H。选票按序分别为1、2、3…8。在作选票统计工作中，有某人的选票，如有1，就在A人的代号下加一票。整个统计过程，应能随时看到每个候选人的得票情况。要求输入方便，程序简练。

解 程序执行一开始，清屏后在指定位置显示出8个候选人的代号A、B、C、……，并在各代号的下方均显示0，表示都是零票。然后再显示输入1～8，这时候只要按下1～8的任一个键，表示某人得一票。连续按键，票数不断变化。若按0键结束，并显示出总票数。

要注意的是，除1～8与0键之外，按其它键则无变化。机器继续等待你重新按键。

程序及示例如下：

程序 9-2

```
  TO XP
  CLEARTEXT MAKE "I 1
  REPEAT 8 [CURSOR :I*4+3 4 PR CHAR
```

```
64+:I MAKE "I :I+1] PR ""
  MAKE "I 1
  REPEAT 8 [MAKE WORD "P :I 0 CURSOR
:I*4+3 6 PR THING WORD "P :I  MAKE "I
:I+1] PR ""
  MAKE "P 0
  AA: CURSOR 4 18 ( PR "INPUT1-8  "END
-0 ) MAKE "T RC
  TEST :T = 0
  IFT CURSOR 4 8 ( PR "ZPS= :P )
PR "" TOPLEVEL
  IFF TEST ANYOF ASCII :T < 48 ASCII
:T > 56
  IFT GO "AA
  IFF MAKE "P :P + 1
  MAKE WORD "P :T (THING WORD "P :T)
+ 1
  MAKE "I 1
  REPEAT 8 [CURSOR  :I*4+3 6 PR THING
WORD "P :I MAKE "I :I+1] PR ""
  GO "AA
END

?XP
A B C D E F G H
2 4 1 2 4 6 1 3
```

ZPS= 23

9-3 成绩统计

若干名学生三门课的考试成绩表已知，按学生学号、语文、数学、外语排列。要求编程统计并打印出符合下列条件之一的学生的学号及各门成绩：

1．三门总分成绩在270分以上者；

2．或前两门成绩均在95分以上者；

3．或第一门100，其它两门在80分以上者，或第二门100，其它两门在80分以上者。

解 将学生学号、语文、数学、外语成绩作为一个子表。此表为若干学生数据总表的一项。这个总表，作为过程的输入。

设置变量A1、A2、A3分别放三门功课的成绩。然后只要按要求条件进行测试，满足条件者，打印输出。否则继续，直到全部学生检查完毕表空为止。

程序如下：

程序 9-3

```
TO KFTJ
  MAKE "N :N ( PR "N= :N )
  MAKE "L :L PK :L
  PR [--------------]
  MAKE "G 0
  HD :L
END
```

```
TO HD :L
  IF :L = [ ] ( PR "N= :G ) STOP
  MAKE "A1 FIRST BF FIRST :L
  MAKE "A2 LAST BL FIRST :L
  MAKE "A3 LAST FIRST :L
  IF ( ANYOF ( :A1 + :A2 + :A3 ) >
270 ALLOF :A1 > 95 :A2 > 95 ( ANYOF
  ( ALLOF :A1 = 100 :A2 > 80  :A3 >
80 ) ( ALLOF :A2 = 100 :A1 > 80 :A3
> 80 ) ) )  PR FIRST :L MAKE "G :G
+1
  HD BF :L
END

TO PK :M
  IF :M = [ ] STOP
  PR FIRST :M
  PK BF :M
END
```

其中 KFTJ 为主过程。在运行之前，用MAKE 命令给出人数(N)，及 N 个学生的数据总表(L)。然后键入 KFTJ 回车，屏幕显示出总人数，原数据；横断线下是符合条件的学生号和成绩，最后一行是符合条件的人数。打印原数据各个表，是由子过程 PK 完成的。检查是否符合条件并打印出符合条件者，是由 HD 过程完成的。

以下是运行示例。原数据 5 人，符合条件的 4 人。

```
?KFTJ
N= 5
101 99 95 91
102 83 85 89
103 91 100 85
104 86 92 99
105 100 85 83
--------------
101 99 95 91
103 91 100 85
104 86 92 99
105 100 85 83
N= 4
```

你还可以将主过程的第一、二行改为：

```
MAKE "N FIRST RQ (PR "N= :N )
MAKE "L RQ PK :L
```

于是，运行时先键入主过程名，回车，再由键盘上输入总人数及数据表。

9-4 及格人数统计

某班学生若干名(设 10 名)，期末考试语文、数学、外语三门课及格者的名单(以班号表示)如下：

语文 1,9,6,8,4,3,7,10

数学 10,5,2,9,1,3,7

外语 8,1,6,7,3,5,4,10,9

编程统计三门课都及格的学生，并打印出学号。

解 假设从语文及格人数中，一个一个取出检查数学、外语是否有他，若有则打印输出学号，否则再取下一个继续检查。LOGO 数据中的表处理，为此提供了十分有趣而且方便的办法。

先设置三个表 L1、L2、L3 分别记三门及格人的班号。以表 L1 为准，逐个检查 L2 与 L3 中是否同时都有。这由子过程 GETN 完成。它调用常用过程 MER？，输出字是否表中的元素。主过程 JGTJ 赋值并启动检查过程。

程序清单及运行结果如下：

程序 9-4

```
TO JGTJ
  MAKE "L1 [1 9 6 8 4 3 7 10] PR :L1
  MAKE "L2 [10 5 2 9 1 3 7] PR :L2
  MAKE "L3 [8 1 6 7 3 5 4 10 9] PR :L3
  MAKE "N 0 PR "------------------
  GETN :L1
END

TO GETN :LL
  IF :LL = [ ] ( PR "N= :N ) STOP
  MAKE "L FIRST :LL
  TEST ALLOF MER? :L :L2 MER? :L :L3
  IFT ( PRINT1 :L "' ' ) MAKE "N :N
+ 1 GETN BF :LL STOP
  IFF GETN BF :LL
```

```
END

TO MER? :W :T
  IF :T = [ ] OP "FALSE
  IF :W = FIRST :T OP "TRUE
  OP MER? :W BF :T
END

? JGTJ
1 9 6 8 4 3 7 10
10 5 2 9 1 3 7
8 1 6 7 3 5 4 10 9
------------------
1 9 3 7 10 N=5
```

可知10人中1、9、3、7、10号5人三门课都及格。

9-5 分数分段统计

将学生的考试成绩以10分为一个分数段，共分成100、90～99、80～89、…10～19、0～9等11个分数段。编程统计每个分数段的人数并打印输出。

解 程序四个过程。二级子过程F1RS给记11个分数段人数的变量赋初值0。变量名采用LOGO复合字，使过程十分简练。它有一个从10、9、…0的11个数为元素的表为输入。二级子过程DUAN统计各段人数，取出数据表中一个分数，属于哪一段，则给其人数加1，它以学生成绩表为输入。三级子过程XIAG由DUAN调用，负责打印出90～

99、80～89、…0～9 这 10 分数段及其人数。另一分数段100 及人数，由 DUAN 完成。

主过程 FDTJ。为示范性，在此随机产生若干个(由输入 N 定)学生成绩。然后调用 FIRS 与 DUAN 过程。

由于巧妙使用 LOGO 复合字作变量名，因而使赋初值、统计各段人数及打印结果，都只使用一个表达式。所以程序模块清晰，作用明了，各模块也十分简练。

程序清单及运行示例如下：

程序 9-5

```
  TO FDTJ :N
  MAKE "L [ ]
  REPEAT :N [MAKE "L SE :L ( RANDOM
101) ]
  PR :L PR "' '
  FIRS [10 9 8 7 6 5 4 3 2 1 0 ]
  DUAN :L
END

TO FIRS :L
  IF :L = [ ] STOP
  MAKE ( WORD "A FIRST :L ) 0
  FIRS BF :L
END

TO DUAN :L
  IF :L = [ ] (PR "100 "' ' :A10)
  XIAG [9 8 7 6 5 4 3 2 1 0] STOP
```

```
  MAKE " I QUOTIENT FIRST :L 10
  MAKE ( WORD "A :I ) ( THING WORD
"A :I ) + 1
  DUAN BF :L
END

TO XIAG :L
  IF :L = [  ] TOPLEVEL
  ( PRINT1 WORD FIRST :L "0--  FIRST
:L "9 "' ')
  (PR THING WORD "A FIRST :L )
  XIAG BF :L
END

? FDTJ 50
82 31 28 71 10 28 69 22 53 84 26
59 64 3 55 88 100 94 12 15 17 61
54 48 73 10 36 93 21 50 16 54 21
85 49 38 27 21 92 53 96 66 0 4 83
24 31 98 45 4
100      1
90—99    1
80—89    5
70—79    2
60—69    4
50—59    7
40—49    3
```

30--39	4
20--29	9
10--19	6
00--09	4

这是个实用程序。在具体使用时，应对主过程稍加修改，删去标题行中的输入N，及第二行命令。将第一行空表换成需统计的数据表赋给变量L。即MAKE "L QR，运行时键入主过程名并回车后，由键盘输入全部成绩，再按回车，则可得出统计结果。

十 组合问题

10-1 由1～4组成的三位数

打印输出由 1～4 组成的没有重复数字的所有三位数。

解 无论百位、十位和个位，都可以是 1～4 四个数字中的一个。所以我们建立三个递归嵌套过程，输入都从 1 变到 4。判定无重复数码则打印输出。

程序及运行结果如下：

程序 10-1

```
TO SS1 :A ;  1-4
  IF :A > 4 STOP
  SS2 1
  SS1 :A + 1
END

TO SS2 :B ;  1-4
  IF :B > 4 STOP
  IF NOT :B = :A SS3 1
  SS2 :B + 1
END

TO SS3 :C ;  1-4
  IF :C > 4 PR "' ' STOP
```

```
  IF ALLOF NOT :C = :B NOT :C = :A
PRINT1 ( WORD :A :B :C ) PRINT1 "' '
  SS3 :C + 1
END

? SS1 1
123  124
132  134
142  143
213  214
231  234
241  243
312  314
321  324
341  342
412  413
421  423
431  432
```

10-2 由 0～9 组成的三位数

由 0～9 个数字可以组成多少字码并不重复的三 位 数。请你编程把它们打印出来。

解 要构成三位数，百位上不能是 0。十位和个位 可以是 0。我们建立三个嵌套的递归过程 SS1、SS2、SS3。SS1 为主过程，控制百位数，它的输入由 1 变到 9。调用 SS2 和

SS3。SS2和SS3控制十位和个位，输入由0变到9。构成一个组合，判定数码是否重复，不重复则打印，否则取下一个再进行判定。

程序及运行的部分结果如下：

程序 10-2

```
TO SS1 :A ;  1-9
  IF :A > 9 STOP
  SS2 0
  SS1 :A + 1
END

TO SS2 :B ;  0-9
  IF :B > 9 STOP
  IF NOT :B = :A SS3 0
  SS2 :B + 1
END

TO SS3 :C ;  0-9
  IF :C > 9 PR "' ' STOP
  IF ALLOF NOT :C = :B NOT :C = :A
PRINT1 ( WORD :A :B :C ) PRINT1 "'  '
  SS3 :C + 1
END

? SS1 1
102 103 104 105 106 107 108 109
120 123 124 125 126 127 128 129
```

```
130 132 134 135 136 137 138 139
140 142 143 145 146 147 148 149
150 152 153 154 156 157 158 159
160 162 163 164 165 167 168 169
170 172 173
```

10-3 无零三位数和等于 13 的数

编程打印出不含 0 的三位数中数字之和为 13 的数。

解 不含 0 的三位数，数码肯定都是由 1 至 9 这九个数字中的一个所组成。我们建立三个相互嵌套的递归过程，对百位、十位和个位，逐一从 1 取至 9 进行组合，然后判断、打印出三位数字之和为 13 者即可。

程序及运行结果如下：

程序 10-3

```
TO SS1 :A ;  1-9
  IF :A > 9 STOP
  SS2 1
  SS1 :A + 1
END

TO SS2 :B ;  1-9
  IF :B > 9 PR "' ' STOP
  SS3 1
  SS2 :B + 1
END

TO SS3 :C ;  1-9
```

```
  IF :C > 9 STOP
  IF :A + :B + :C = 13 PRINT1 ( WORD
:A :B :C ) PRINT1 "' '
  SS3 :C + 1
END

? SS1 1
139 148 157 166 175 184 193
229 238 247 256 265 274 283 292
319 328 337 346 355 364 373 382 391
418 427 436 445 454 463 472 481
517 526 535 544 553 562 571
616 625 634 643 652 661
715 724 733 742 751
814 823 832 841
913 922 931
```

10-4 用一元、二元、五元凑十元的方法

用一元、二元、五元面值的美元凑十美元，共有多少种方法？每种方法中各面值美元有几张？编程解决这个问题。

解 凑十美元，用五元的张数可以是0张、1张、2张；用二元的张数范围是 0～5 张；用 一 元 的张数范围是 0～10 张。采用穷举法，把所有可能取的情况都试验一遍，判断能否凑成十美元。若满足，就打印输出一组。不行，再继续判断另一种。

法 1 用 REPEAT 命令，嵌套三层编程。各层重复次

数分别是 3、6、11 次。程序如下。程序由一个过程 QQ1 完成。

程序 10-4-1

```
TO QQ1
  PR ( SE "5$ "' ' "2$ "' ' "1$ )
  MAKE "T 0 MAKE "A 0
  REPEAT 3 [MAKE "B 0 REPEAT 6 [MAKE
"C 0 REPEAT 11 [ IF :A+:B+:C=10 PR
  (SE :A/5 "' ' :B/2 "' ' :C) MAKE "T
:T+1 MAKE "C :C+1 ELSE MAKE "C
:C+1] MAKE "B :B+2 ] MAKE "A :A+5]
  (PR "T= :T)
END

?QQ1
5$          2$          1$
0.          0.          10
0.          1.          8
0.          2.          6
0.          3.          4
0.          4.          2
0.          5.          0
1.          0.          5
1.          1.          3
1.          2.          1
2.          0.          0
T=10
```

可见，共有 10 种凑法，其中每种凑法各面值的张数即上表中的数。

法 2 三重循环用三个递归过程嵌套调用也可以。过程 QQA、QQB、QQC 嵌套调用，它们的输入都从 0 到 10(即 0 元到 10 元)，但面值不同，即步长不同。分别加 5、2、1。最后由 QQC 判定 A + B + C = 10 的一种组合，即打印输出并给记种数变量 T 加 1。设立主过程 QQK，给 T 赋初值，打印表头及最后总种数。

程序及运行结果如下：

程序 10-4-2

```
TO QQK
  MAKE "T 0
  ( PR "5$ "' ' "2$ "' ' "1$ )
  QQA 0
  ( PR "T=  :T )
END

TO QQA :A
  IF :A > 10 STOP
  QQB 0
  QQA :A + 5
END

TO QQB :B
  IF :B > 10 STOP
  QQC 0
  QQB :B + 2
```

```
END

TO QQC :C
  IF :C > 10 STOP
  IF :A + :B + :C = 10 (PR :A / 5 "' '
:B / 2 "' ' :C ) MAKE "T :T+ 1 QQC :C
+ 1 ELSE QQC :C + 1
END

? QQK
5$        2$        1$
0.        0.        10
0.        1.        8
0.        2.        6
0.        3.        4
0.        4.        2
0.        5.        0
1.        0.        5
1.        1.        3
1.        2.        1
2.        0.        0
T=10
```

10-5 由 a、b、c 组成 n 位数的种数

由 a、b、c 三个不同的非零数字组成 n 位数，要求不出现

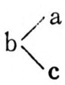
两个 a 相邻，也不出现两个 b 相邻。问共有多少个 n 位数（n=1、2、3…）。

解 分析推导一下。当 n=1 时，a、b、c 各是个 1 位数。所以共有 3 个 1 位数。

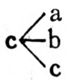
当 n=2 时，由 a、b、c 可这样组成 2 位数：



（ab、ac）



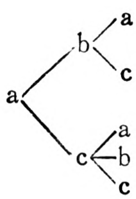
（ba、bc）



（ca、cb、cc）

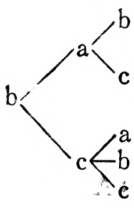
共有 7 个 2 位数。

当 n=3 时，由 a、b、c 可这样组成 3 位数：



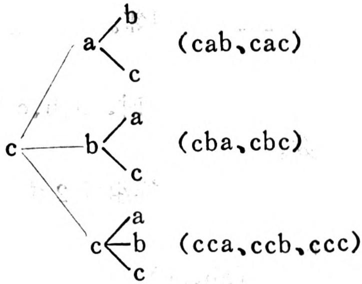
（aba、abc）

（aca、acb、acc）



（bab、bac）

（bcb、bca、bcc）



共有 17 个 3 位数。

由此可得递归方程，n 位数的个数

$$a_n = 2 \cdot a_{n-1} + a_{n-2}$$

初值$a_0 = 1$，$a_1 = 3$。

于是不难编程如下。主过程 ABC 输入要求的位数 n。子过程 SZS，两个输入，从 1 到 N。递归 N 次，求出数的个数 A。

程序及运行示例如下：

程序 10-5

```
TO ABC :N
  MAKE "A 1
  MAKE "B 3
  SZS 1 :N
  ( PR "N= :N "A= :A )
END

TO SZS :I :N
  IF :I > :N STOP
  MAKE "A 2 * :B + :A
```

```
  MAKE "C :B
  MAKE "B :A
  MAKE "A :C
  SZS :I + 1 :N
END

? ABC 1
N= 1  A= 3
? ABC 2
N= 2  A= 7
? ABC 3
N= 3  A= 17
? ABC 4
N= 4  A= 41
? ABC 9
N= 9  A= 3363
```

十一　逻辑推理问题

11-1　好人好事是谁做的？

有 A、B、C、D、E 五人，其中一人做了件好事，但不愿说出。当问他们是谁做了好事时，回答是：A说“不是我”；B 说“是 C 做的”；C 说“是 D 做的”；D说“别听 C 胡说”；E 说“反正不是 D 和我”。若能知道五人中四人说的是真话，一个人说的是假话。试编程序判断究竟是谁做了好事。

解法 1　假设 K 代表做好事的人，K 可取 A、B、C、D、E 五个不同的值，一个一个地测试，来判断他们说话的真假。五人回答的话用以下表达式表示：

```
A 说：NOT :K="A
B 说：:K="C
C 说：:K="D
D 说：NOT :K="D
E 说：ALLOF NOT :K="D NOT :K="E
```

用 I 来记说真话的人　只要以上各表达式之一成立，就给 I 加 1，最后看 I 是否等于 4。做好事的人就是 I 等于 4 时 K 所取值的那一个人。

程序只有一个过程 S1，带一个输入。将 A、B、C、D、E 作元素按表输入，或将 ABCDE 按字输入都行。

程序及运行结果如下：

程序 11-1-1

```
TO S1 :L ; [ A B C D E]OR"ABCDE
  MAKE "K FIRST :L
  MAKE "I 0
  IF NOT :K = "A MAKE "I :I + 1
  IF :K = "C MAKE "I :I + 1
  IF :K = "D MAKE "I :I + 1
  IF NOT :K = "D MAKE "I :I + 1
  IF ALLOF NOT :K = "D NOT :K = "E
MAKE "I :I + 1
  IF :I = 4 ( PR "IS :K ) STOP
  S1 BF :L
END

? S1 [A B C D E]
IS C

? S1 "ABCDE
IS C
```

两种输入方法其运行结果都表示，做好事的人是 C。

解法 2 将五人说话的真或假的逻辑值，转换成数字1或0，分别存入变量 AS、BS、CS、DS、ES 中，让 K 逐一取 A、B、C、D 或 E 去进行检查。当五人回答的变量和为 4 时，K 所取得的值，就是做好事的人。这个程序两个过程，子过程 ZJ 完成逻辑值的转换。S2 为主过程。其输入为字"ABCDE 或表[A B C D E]亦可。如键入 S2 "ABCDE 回车，结果显示 C。

程序及运行结果如下：

程序 11-1-2

```
TO S2 :W ; "ABCDE OR [A B C D E]
  MAKE "K FIRST :W
  MAKE "AS ZJ [NOT :K="A]
  MAKE "BS ZJ [:K="C]
  MAKE "CS ZJ [:K="D]
  MAKE "DS ZJ [NOT :K="D]
  MAKE "ES ZJ [ALLOF NOT :K="D NOT
:K="E]
  IF :AS + :BS + :CS + :DS + :ES =
4 PR :K STOP
  S2 BF :W
END

TO ZJ :L
  IF ( RUN :L ) = "TRUE OP 1
  OP 0
END

? S2 "ABCDE
C
```

打印出 C，说明做好事的人是 C。

11-2　对竞赛名次的预测

亚运会上三人对一个竞赛项目结果进行预测。他们预测的情况是：

甲说：运动员 A 得第一名；运动员 B 得第三名。

乙说：运动员 C 得第一名；运动员 D 得第四名。

丙说：运动员 D 得第一名；运动员 B 得第三名。

比赛结束，竞赛结果表明，他们每个人都只说对了一半，说错了一半。试请你编程排出名次。

解 四个运动员 A、B、C、D 的名次分别用 MA、MB、MC、MD 来表示。则 MA=1，MB=3 便表示 A 得第一名，B 得第三名。这两个式子要一对一错，即要逻辑运算命令 ANYOF(ALLOY :MA=1 NOT :MB=3) (ALLOY NOT :MA=1 :MB=3)成立。同样的道理，其它两人的预测也有相似的表示。这样，我们分别让四个运动员取 1 至 4 名，当某一次三人说话的条件都满足的时候，名次也就排定了。

程序由三个过程组成。主过程 AA，带一个输入，它是使运动员 A 的名次由 1 到 4 的递归过程，并调用子过程BB。子过程 BB 类似于主过程 AA，它使运动员 B 的名次分别从 1 到 4，并调用子过程 CC。它是程序的关键。它除了给运动员C及D取不同的名次外，还完成判断及打印输出运行结果。

程序及运行结果如下：

程序 11-2

```
TO AA :N1 ;  1-4
  IF :N1 > 4 STOP
  MAKE "MA :N1
  BB 1
  AA :N1 + 1
END
```

```
TO BB :N2
  IF :N2 > 4 STOP
  MAKE "MB :N2
  IF NOT :MB =:MA CC 1
  BB :N2 + 1
END

TO CC :N3
  IF :N3 > 4 STOP
  MAKE "MC :N3
  IF ( ANYOF :MC = :MA :MC = :MB )
CC :N3 + 1
  MAKE "MD 10 - :MA - :MB - :MC
  IF ( ALLOF (ANYOF ( ALLOF :MA =
1 NOT :MB = 3 ) ( ALLOF NOT :MA = 1
:MB = 3 ) ) (ANYOF ( ALLOF :MC =
1 NOT :MD = 4 ) ( ALLOF NOT :MC = 1
:MD = 4 ) ) ( ANYOF ( ALLOF :MD =
1 NOT :MB = 3 ) ( ALLOF NOT :MD = 1
:MB = 3 ) ) ) THEN ( PR [A IS] :MA)
  ( PR [B IS] :MB ) ( PR [C IS] :MC )
( PR [D IS] :MD ) TOPLEVEL
  CC :N3 + 1
END

?AA 1
A IS 4
```

B IS 3

C IS 1

D IS 2

结果表明 C 第一，D 第二，B 第三，A 第四。

11-3 湖泊大小次序问题

有四个学生对我国四大淡水湖的排列次序作如下回答：

甲说：洞庭湖最大，洪泽湖最小，鄱阳湖第三；

乙说：洞庭湖最小，洪泽湖最大，鄱阳湖第二，太湖第三；

丙说：洞庭湖第三，洪泽湖最小；

丁说：洪泽湖第二，鄱阳湖最大，太湖最小。

他们四人的回答，每个人都只说对了一个，请编程排列出正确的次序。

解 编程思路仍然是分别让四个湖泊从 1 到 4 取不同的次序进行组合，然后对每一种组合，判断学生回答是否满足各人只说对一个的条件。如是，该组合名次就是正确的。如非，则重新组合。

但问题是，如果再用逻辑运算命令 ALLOF、ANYOF 来判断，对三个、甚至四个回答中只有一个是正确的，那就太麻烦了，几乎不可能。故在此我们使用一点技巧，改逻辑运算为数字运算，来判断条件的真假。为此，仍以 D、H、B、T 四变量的值表示洞庭湖、洪泽湖、鄱阳湖及太湖的次序。引进绝对值函数过程 ABS 及符号函数过程 SGN 。那么，对甲来说，他回答的三个次序，若有一个正确，关系式

3－(SGN ABS :D－1)－(SGN ABS :H－4)－

(SGN ABS :B－3)

的值就应等于1，否则就不等于1。其中:D－1就表示洞庭湖最大，:H－4表示洪泽湖最小。:B－3表示鄱阳湖第三。同理，对于其它三人的回答，也有类似关系式。只要逐一判断，对某个名次的组合，四人回答的条件同时满足，即是正确的名次。

程序由五个过程组成。子过程ABS、SGN已如前所述。主过程DDD，带一个输入，使变量D从1到4取值，并调用子过程HHH。子过程HHH使变量H也从1到4取值，并调用BBB子过程。子过程BBB是关键，它完成判断及打印结果的全部工作。

程序及运行结果如下：

程序 11-3

```
TO DDD :N
  IF :N > 4 STOP
  MAKE "D :N
  HHH 1
  DDD :N + 1
END

TO HHH :N
  IF :N > 4 STOP
  MAKE "H :N
  IF :H = :D HHH :N + 1
  BBB 1
  HHH :N + 1
```

```
END

TO BBB :N
  IF :N > 4 STOP
  MAKE "B :N
  IF ANYOF :B = :D :B = :H GO 10
  MAKE "T 10 - :D - :H - :B
  MAKE "Z 3- ( SGN ABS :D - 1 )
-( SGN ABS :H - 4 ) - ( SGN ABS :B
-3)
  IF NOT :Z = 1 GO 10
  MAKE "Z 4 - ( SGN ABS :D -4 )
-( SGN ABS :H -1 ) - ( SGN ABS :B
-2) - ( SGN ABS :T -3 )
  IF NOT :Z = 1 GO 10
  MAKE "Z 2 - (SGN ABS :D - 3 )
-( SGN ABS :H - 4 )
  IF NOT :Z = 1 GO 10
  MAKE "Z 3 - ( SGN ABS :H -2 )
-(SGN ABS :B -1 ) - ( SGN ABS :T
-4 )
  IF NOT :Z = 1 GO 10
  ( PR [DH IS] :D ) ( PR [HH IS] :H)
  ( PR [BH  IS] :B ) ( PR [TH IS] :T )
  TOPLEVEL
  10: BBB :N + 1
```

```
END

TO ABS :X
  IF :X < 0 OP - :X
  OP :X
END

TO SGN :X
  IF :X < 0 OP - 1
  IF :X = 0 OP 0
  OP 1
END

?DDD 1

DH IS 2
HH IS 4
BH IS 1
TH IS 3
```

运行结果表明鄱阳湖最大，洞庭湖第二，太湖第三，洪泽湖最小。

11-4 破　　案

LOGO巧破案。设有 x_1、x_2、x_3、x_4、x_5 五人可能参与了一件凶杀案。关于到底哪些人参与此案没有直接结论，但却有以下几条可靠线索：

(1) 如果 x_1 参与作案，x_2 也参与作案。

（2）x_2 与 x_3 只有一人参与此案

（3）x_3 和 x_4 二人或者同时参与此案，或者都与本案无关。

（4）x_4 和 x_5 二人中，至少一人参与此案。

（5）如果 x_5 参与作案，x_1 与 x_4 也一定参与此案。

请问到底谁是罪犯？

解 若用 1 表示作案，0 表示未作案，对五人分别用 0、1 来进行组合判断。满足所提供的五条线索的一组组合，便是作案与未作案的答案。以第一条线索为例，把它写成判断条件就是值 $x_1=x_2$，即同时作案为 1，都不作案为 0，取值应相同。第二条线索则可写成 $x_2+x_3=1$。即是 x_2 作案，取 1，则 x_3 就未作案，取 0。反过来，若 x_3 作案，取 1，则 x_2 就应为 0。余类推。

于是可编制如下的六个过程：A、B、C、D、E 及 SGN，来进行判断。其中 A 为主过程。前五个过程的递归嵌套相当于 5 重循环，使其输入都从 0 到 1，然后逐个对 5 条线索进行判断。最后一个过程是符号函数过程 SGN。

程序及运行结果如下：

程序 11-4

```
TO A :X1
  IF :X1 > 1 STOP
  B 0
  A :X1 + 1
END

TO B :X2
```

```
  IF :X2 > 1 STOP
  IF :X1 = :X2 C 0
  B :X2 + 1
END

TO C :X3
  IF :X3 > 1 STOP
  IF :X2 + :X3 = 1 D 0
  C :X3 + 1
END

TO D :X4
  IF :X4 > 1 STOP
  IF :X3 * :X4 + ( 1 - :X3 ) * ( 1
- :X4 ) = 1 E 0
  D :X4 + 1
END

TO E :X5
  IF :X5 > 1 STOP
  IF NOT ( SGN :X4 + :X5 ) = 1 E :X5
+ 1 STOP
  IF :X4 * :X5 = :X1 ( PR "'X1 ' "'
X2 ' "'X3 ' "'X4 ' "'X5 ' ) ( PR :X1
"' ' :X2 "' ' :X3 "' ' :X4 "' ' :X5 )
STOP
  E :X5 + 1
```

```
END

?A 0
X1 X2 X3 X4 X5
 0  0  1  1  0
```

由结果可知，罪犯是 x_3 与 x_4 二人，其余三人未参与作案。

十二 近似计算应用

12-1 求n次多项式的值

编程计算多项式

$$f(x)=a_nx^n+a_{n-1}x^{n-1}+\cdots+a_2x^2+a_1x+a_0$$

的值。已知n、x及a_n、$a_{n-1}\cdots a_1$、a_0等。

解 在计算机上求n次代数多项式的值，一般采用秦九韶法。外国人称作霍纳(HONER)法。因为

$$f(x)=a_nx^n+a_{n-1}x^{n-1}+\cdots+a_2x^2+a_1x+a_0$$
$$=(\cdots((0\cdot x+a_n)x+a_{n-1})x+\cdots+a_1)x+a_0$$

可见，只要将一乘一加重复多次就行了。

在此，把各项系数作为表的元素，缺项系数以0补齐，将该表作为主过程的第一个输入。x的值，作为第二个输入。子过程除上述两个输入之外，第三个输入从0开始，用来记函数的值。

程序的两个过程如下。其中QJS为主过程。

程序 12-1

```
TO QJS :A :X
  IF :A = [ ] OP 0
  OP QJS1 :A :X 0
END

TO QJS1 :A :X :Y
```

```
  IF :A = [ ] OP :Y
  OP QJS1 BF :A :X :Y * :X + FIRST
  :A
END
```

譬如，欲求 $f(x)=x^3+4x^2+7x+8$ 在 $x=0.5$ 处的值。则键入的内容及运行结果如下：

```
?PR QJS [1 4 7 8] .5
12.625
```

12-2　二分法求方程近似实根

用二分法求方程 $f(x)=0$ 的近似实根。

解　求方程近似根的方法很多，二分法是一种常用的方法。如果方程在区间$[x_1,x_2]$必有一个实根，且只有一个实根，就可以用二分法求这个根。

原理是这样的：由于方程在区间$[x_1,x_2]$只有一个根，所以 $f(x_1)$和 $f(x_2)$异号，即 $f(x_1)\cdot f(x_2)<0$。把区间二等分，求中点 $x=\frac{x_1+x_2}{2}$，然后判断 $f(x_1)$和 $f(x)$（或者 $f(x)$ 和 $f(x_2)$）的符号。如果 $f(x_1)\cdot f(x)<0$，则根在区间$[x_1,x]$内，这时用 x 代替 x_2；反之，如果 $f(x_1)\cdot f(x)>0$，则根在区间$[x,x_2]$内，这时用 x 代替 x_1。然后，把根所在的区间再二等分，重复上述过程。这样一次次二等分根所在的区间，则区间越来越小。当区间两个端点的差的绝对值小于预先确定的精确度时，其任一端点就可以作为方程的近似实根。

法1　主过程 WTO，子过程 FUN 及 ABS。主过程带

两个输入，即开始时的区间端点。过程中变量 C 为中点坐标；YA、YC 是左端点 A 与中点函数值。当端点差的绝对值小于 10^{-6}(即程序中的 9.9999N7)时，运行结束，打印结果。子过程 FUN 求函数值。

程序及运行结果如下。它是求方程

$$3x^3-8x^2+7x-16=0$$

在区间[2,3]的近似实根。

程序 12-2-1

```
TO WTO :A :B
  MAKE "C ( :A + :B ) / 2
  MAKE "YA FUN :A
  MAKE "YC FUN :C
  IF :YA * :YC < 0 MAKE "B :C GO "AA
  MAKE "A :C
  AA: IF ABS :A - :B < 9.9999N7 ( PR
"X= :C ) STOP
  WTO :A :B
END

TO FUN :X
  OP 3 * :X * :X * :X - 8 * :X * :X
+ 7 * :X - 16
END

?WTO 2 3
X= 2.56705
```

法 2 为了具有通用性，我们建立主过程 WTO1，把方程左端的函数式，以表作为过程的第一个输入。其余的两个端点不变。这样省去 FUN 过程。它只带一个子过程 ABS。

程序及运行结果如下。它既可以求法 1 中的方程的近似实根，也可以求方程

$$x-\sin x-1=0$$

在区间[0,π]的近似实根。而程序不加任何变动。通用性强。

程序 12-2-2

```
TO WTO1 :F :A :B
  MAKE "C ( :A + :B ) / 2
  MAKE "X :A
  MAKE "YA RUN :F
  MAKE "X :C
  MAKE "YC RUN :F
  IF :YA * :YC < 0 MAKE "B :C ELSE
MAKE "A :C
  IF ABS :A - :B < 9.99991N7 ( PR"X
= :C ) STOP
  WTO1 :F :A :B
END

?WTO1 [3*:X*:X*:X-8*:X*:X+7*:X
  -16] 2 3
X= 2.56705

?WTO1 [:X-(SIN :X*180/3.14159)-1] 0
  3.14159
```

X= 1.93455

12-3 优选法求方程近似实根

用优选法求方程 f(x)=0 的近似实根。

解 用优选法求方程 f(x)=0 的近似实根，可以迅速缩短区间长度,从而迅速达到预定精度的要求。比二分法优越。参见 12-2 题。

如求 $3x^3-8x^2+7x-16=0$ 在区间 [2,3] 的近似实根。程序及运行结果如下：

程序 12-3-1

```
TO YOU :A :B
  MAKE "C :A + .618 * ( :B - :A )
  MAKE "YA FUN :A
  MAKE "YC FUN :C
  IF :YA * :YC < 0 MAKE "B :C GO "AA
  MAKE "A :C
  AA: IF ABS :A - :B < 9.99995N7 ( PR
"X= :C ) STOP
  YOU :A :B
END

TO FUN :X
  OP 3 * :X * :X * :X - 8 * :X * :X
+ 7 * :X - 16
```

```
END

?YOU 2 3
X= 2.56705
```

如下程序更为通用。

程序 12-3-2

```
TO YOU1 :F :A :B
  MAKE "C :A + .618 * ( :B - :A )
  MAKE "X :A
  MAKE "YA RUN :F
  MAKE "X :C
  MAKE "YC RUN :F
  IF :YA * :YC < 0 MAKE "B :C ELSE
MAKE "A :C
  IF ABS :A - :B < 9.99995N7 ( PR "X
= :C ) STOP
  YOU1 :F :A :B
END

?YOU1 [3*:X*:X*:X-8*:X*:X+7*:X-16]
  2 3
X= 2.56705
```

12-4 迭代法求方程近似实根

用迭代法求方程 f(x) = 0 实根的近似值。

解 对不容易用初等数学方法求解的代数方程或超越方

程，常可用迭代法求它的近似根。如欲求五次方程：

$$x^5 - 4x - 2 = 0$$

的近似根，具体方法步骤是：

首先将方程变形，写成 $x = h(x)$ 的形式。即

$$x = \frac{1}{4}x^5 - \frac{1}{2}$$

然后对方程的根作粗略估计，确定一个 x 的初始近似值，将它代入上式右端，算出一个新的 x 值。把新的 x 又代入上式右端，又算出一个 x 的新值。如此过程（即称迭代过程）一直继续到相邻两次求得的 x 的值充分接近，满足预定的精度为止。

于是，程序只要能完成函数值的计算，迭代及终止规则，给初值及打印最终结果就行了。

主过程 JDF1，给 x 初始近似值 1。打印结果并调用子过程 ITE，ITE 完成变量的传递即迭代过程。当两次根的近似值之差小于 1.N6（10^{-6}）时迭代停止。子过程 FUN 用来计算函数值。它还调用绝对值函数常用过程。

程序及运行结果如下：

程序 12-4-1

```
TO JDF1
  MAKE "X 1
  ( PR "X= ITE )
END

TO ABS :X
  IF :X < 0 OP - :X
  OP :X
```

```
END

TO ITE
  MAKE "Y FUN :X
  IF ABS :Y - :X < 9.99997N7 OP :Y
  MAKE "X :Y
  OP ITE
END

TO FUN :X
  OP :X * :X * :X * :X * :X / 4 - .5
END

?JDF1
X= -.508499
```

以上程序，只能解固定的一个方程，不大通用。下面将主过程改为JDF，且将原方程变形后右端的函数式作为该过程的第一个输入，以表给出。x的初始近似值作为第二个输入。省去原程序中的两个子过程FUN及ITF。留用ABS子过程。这样，灵活方便，通用性强。

程序及运行结果如下：

程序 12-4-2

```
TO JDF :F :X
  MAKE "Y RUN :F
  IF ABS ( :Y - :X ) < 9.99995N7 ( PR
"X= :Y ) STOP
  JDF :F :Y
```

```
END
?JDF [:X*:X*:X*:X*:X/4-.5] 1
X= -.508499
```

12-5 梯形法求定积分的值

编程用梯形公式法求定积分 $P=\int_a^b f(x)dx$ 的值。

解 求积分 $P=\int_a^b f(x)dx$，可以看作是求由函数曲线 $f(x)$、x 轴以及 $x=a$ 和 $x=b$ 四条线所围成的图形的面积。如图 12-1。

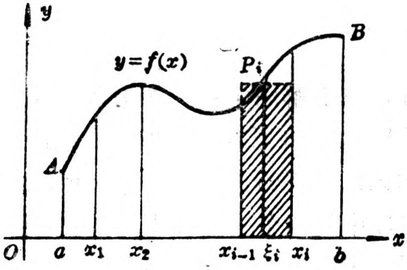


图 12-1

用梯形法求$\int_a^b f(x)dx$，是把[a,b]区间分成 n 等分，每个区间宽度为 H。$H=\frac{b-a}{n}$。每个区间端点的横坐标分别为 $a, x_1=a+H, x_2=a+2H, \cdots x_n=a+n\cdot H$ 及 b。而每两

个相邻分点的函数值，可以看作是小梯形的上下底。区间宽度H作为小梯形的高。于是每个小梯形的面积便为$(f(x_{n-1})+f(x_n))\cdot\frac{H}{2}$，n个小梯形面积之和就可以近似地作为函数f(x)在[a,b]区间的积分值。当n越大，其误差越小。从而得出：

$$P=\int_a^b f(x)dx=\frac{H}{2}(y_0+y_1)+\frac{H}{2}(y_1+y_2)$$

$$+\frac{H}{2}(y_2+y_3)+\cdots+\frac{H}{2}(y_{n-1}+y_n)$$

$$=\frac{H}{2}(y_0+2y_1+2y_2+\cdots+2y_{n-1}+2y_n-y_n)$$

$$=\frac{H}{2}[(y_0-y_n)+2(y_1+y_2+y_3+\cdots+y_n)]$$

$$=H\cdot[(y_0-y_n)/2+(y_1+y_2+y_3+\cdots+y_n)]$$

这样变成了一个累加问题。只要能计算出区间长度H，及被积函数在各分点的值，问题就解决了。

程序由三个过程组成。主过程QXF，计算区间长度及打印积分结果。它带三个输入，两个端点及等分数。子过程FUN计算函数值。子过程SUM求被积函数累加和。终止条件是当N=1时，只输出两个区间端点函数值和的一半。下边是求$\int_0^1\frac{dx}{1+x^2}$的值的程序及等分数分别为10、50、100的运行结果：

程序12-5

```
TO QXF :A :B :N
```

```
  MAKE "H ( :B - :A ) / :N
  PR :H*SUM :N
END
TO SUM :N
  MAKE "Y 0
  AA: IF  :N = 1 OP ( ( FUN :A ) +
FUN :B ) / 2 + :Y
  MAKE "Y :Y + FUN ( :A + ( :N - 1 )
 * :H )
  MAKE "N :N - 1
  GO "AA
END
TO FUN :X
  OP 1 / ( 1 + :X * :X )
END
?QXF 0 1 10
.784981
?QXF 0 1 50
.78538
?QXF 0 1 100
.78539
```

又如欲求$\int_0^1 \frac{\sin x}{x} dx$,将[0,1]区间分别等分为 10、100、1000 时的运行结果及函数式过程修改如下:

```
TO FUN :X
  IF :X = 0 OP 1
  OP ( SIN :X * 180 / 3.14159 ) / :X
END

?QXF 0 1 10
.945806

?QXF 0 1 100
.946051

?QXF 0 1 1000
.946018
```

12-6 优选法求函数极大点

用优选法求函数 f(x) 的极大点。

解 极值问题也是数值计算中经常遇到的问题。用优选法求极大点，原理是这样的：

假设函数 f(x) 在区间[a,b]中是单峰性函数，且有唯一极大点 x_m。而 f(x)在 x_m 左上升，在 x_m 右下降。

通常用缩短区间的办法求出 f(x) 在区间[a,b]的极大点。

在[a,b]内任选两点 c、d(c<d)。若 f(c)<f(d)，则极大点必在区间[c,b]内，从而将区间[a,b]缩短为[c,b]。反之，若 f(c)≥f(d)，则极大点必在区间[a,d]内，于是把区间[a,b]缩短为[a,d]。重复上述过程，最终使所缩短的区间充分小，小到满足预先确定的某一精确度为止。这时，区间

任一端点就可作为函数的极大点。

问题是如何选取 c、d，才能使区间缩短最快。这就是所用的优选法，或称 0.618 法。

首先在区间 [a,b] 长的 0.618 处，取一点作为 d，然后对[a,b]的中点取 d 点的对称点作为 c。即下式

$$\begin{cases} d = a + 0.618(b - a) \\ c = a + b - d \end{cases}$$

在缩短了的区间内，重复上述步骤，使区间长迅速达到预定精度要求。

法 1 程序用四个过程。主过程 MAX，有两个输入，为初始的区间端点。它先求出 D 及 C 的值，然后调用子过程 SOL 并负责打印输出信息。子过程 SOL 用以判断新区间端点函数值的大小及区间端点值的传递。子过程 FUN 输出函数值，还有绝对值函数过程也作为子过程。

如用优选法求函数 $f(x) = \sqrt{3x^2 - x^3}$ 分别在区间[0,1]、[0,2]、[0,3]内的极大点，程序及运行结果如下：

程序 12-6-1

```
TO MAX :A :B
  MAKE "D :A + .618 * ( :B - :A )
  MAKE "C :A + :B - :D
   ( PRINT1 "( :A ", :B") "' ' "MAX "' '
"IS "' ' )
  SOL
END

TO SOL
  IF FUN :C < FUN :D MAKE "A :C ELSE
```

```
MAKE "B :D
  IF ABS ( :A - :B ) < 9.99997N7 PR
  :A STOP
  MAKE "D :A + .618 * ( :B - :A )
  MAKE "C :A + :B - :D
  SOL
END

TO FUN :X
  OP SQRT 3 * :X * :X - :X * :X * :X
END

?MAX 0 1
(0,1) MAX IS .999999

?MAX 0 2
(0,2) MAX IS 1.99931

?MAX 0 3
(0,3) MAX IS 1.99882
```

如同 12-4 题一样，为提高程序通用性，修改如下（主过程 MAX1，子过程 SOL1）：

程序 12-6-2

```
TO MAX1 :F :A :B
  MAKE "D :A + .618 * ( :B - :A )
  MAKE "X :D
  MAKE "YD RUN :F
  MAKE "C :A + :B - :D
```

```
  MAKE "X :C
  MAKE "YC RUN :F
  ( PRINT1 "( :A ", :B ")"' ' "MAX "' '
"IS "' ' )
  SOL1
END

TO SOL1
  IF :YC < :YD MAKE "A :C ELSE MAKE
"B :D
  IF ABS ( :A - :B ) < 9.99997N7 PR
:A STOP
  MAKE "D :A + .618 * ( :B - :A )
  MAKE "X :D
  MAKE "YD RUN :F
  MAKE "C :A + :B - :D
  MAKE "X :C
  MAKE "YC RUN :F
  SOL1
END

?MAX1 [3*:X*:X-:X*:X*:X] 0 1
(0,1) MAX IS .999999

?MAX1 [3*:X*:X-:X*:X*:X] 0 2
(0,2) MAX IS 1.99931
```

十三　杂　题

13-1　改变矩阵形式

输入一个如下的 3×3 矩阵：

1	2	3
4	5	6
7	8	9

使元素位置改变后，按如下的一些形式输出：

```
9  8  7        7  8  9        3  2  1
6  5  4        4  5  6        6  5  4
3  2  1,       1  2  3,       9  8  7。
```

解　LOGO 中没有数组的概念，但表处理的功能却很强。为了实现矩阵的输出，我们把输入的 3×3 矩阵作为一个总表，各行元素作一个子表。这样利用表处理命令，即可按需要输出。

程序共由六个过程组成。CC 为主控过程，它带的输入是[[1 2 3][4 5 6][7 8 9]]。其余子过程分别完成输入矩阵的打印及要求变形之后的矩阵的打印。具体的讲，PP 打印第一个矩阵；AA、BB 打印第二个矩阵；DD 打印第三个矩阵；EE 打印最后一个矩阵。

程序及运行结果如下：

程序 13-1

```
TO CC :L; [ [1 2 3] [4 5 6] [7 8 9] ]
  PP :L PR "' '
  AA :L PR "' '
  DD :L PR "' '
  EE :L PR "' '
END

TO PP :L
  IF :L = [ ] STOP
  PR FIRST :L
  PP BF :L
END

TO AA :L
  IF :L = [ ] STOP
  PR BB LAST :L
  AA BL :L
END

TO BB :L
  IF :L = [ ] OP [ ]
  OP SE LAST :L BB BL :L
END

TO DD :L
  IF :L = [ ] STOP
  PR LAST :L
```

```
  DD BL :L
END

TO EE :L
  IF :L = [ ] STOP
  PR BB FIRST :L
  EE BF :L
END

?CC [[1 2 3] [4 5 6] [7 8 9] ]
1  2  3
4  5  6
7  8  9

9  8  7
6  5  4
3  2  1

7  8  9
4  5  6
1  2  3

3  2  1
6  5  4
9  8  7
```

13-2 把一个数分成二位、三位、四位数

由1、3、8、5、7、9、6、4、2九个数彼此首尾相接排成一圈。

请你按次序把它们分成一个两位数、三位数和四位数。使二位数乘以三位数等于四位数。可以从任意一数开始。

解 对这一圈数，我们从任意一个数开始，组成一个二位数、三位数和四位数，分别赋给变量 B、C、D。然后进行测试。只要 B×C 等于 D，就打印输出。否则，利用字操作命令，将第一个数码移至末尾，重新组成二位、三位和四位数再试。

定义主过程 FENS。它带两个输入，第一个是由所给数码组成的字。第二个输入控制递归调用次数。该程序还有两个子过程。分别为 MID$ 和 PICK 字函数过程。前者输出指定的子字。后者输出字中指定的字符。

程序及运行结果如下：

程序 13-2-1

```
TO FENS :A$ :T; "138579642 1
  IF :T > 9 STOP
  MAKE "B MID$ :A$ 1 2
  MAKE "C MID$ :A$ 3 3
  MAKE "D MID$ :A$ 6 4
  IF :B * :C = :D ( PR :B "* :C "= :D )
  FENS ( WORD BF :A$ FIRST :A$ ) :T
  + 1
END


?FENS "138579642 1
42 * 138 = 5796
```

其实由于运行结果是唯一的，可以删去控制次数的输

入。且只要有打印输出，即告结束。原程序修改如下：

程序 13-2-2

```
TO FENS1 :A$ ;"138579642
  MAKE "B MID$ :A$ 1 2
  MAKE "C MID$ :A$ 3 3
  MAKE "D MID$ :A$ 6 4
  IF :B * :C = :D ( PR :B "* :C "=
 :D ) STOP
  FENS1 ( WORD BF :A$ FIRST :A$ )
END

?FENS1 "138579642
42 * 138 = 5796
```

可见，运行结果是一样的。

13-3 圆盘上找数

如图 13-1 所示的 20 个数彼此首尾相接排成一圈。从任

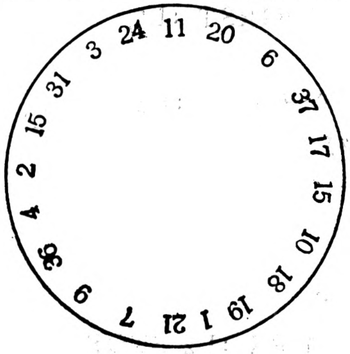


图 13-1

意一个数开始，把它们按次序四个四个数相加，找出其中和为最大及最小的四个数，并同时打印出其和与这两组数。

解 原数据放表 L1 中，四个四个求和之后放变量 L2 中。因为首尾相接求和，故变量 L1 末尾多加第一、二、三项三个元素。为了求和，定义子过程 SUM，对四个四个元素求和。主过程中的其它变量 M、I、T、J 及子过程 PAI，类似于 8-2 题程序，它们负责比较并输出和中最大的一个及最小的一个。程序还调用子过程 PICK。主过程名为YPZS。

程序及运行结果如下：

程序 13-3

```
TO YPZS
  PR "'
  MAKE "L1 [19 1 21 7 9 36 4 2 15 3
  1 3 24 11 20 6 37 17 15 10 18 19 1 21]
  PR :L1 PR "'
  MAKE "L2 [ ]
  SUM 1 PR "'
  MAKE "M FIRST :L2
  MAKE "I 1
  MAKE "T FIRST :L2
  MAKE "J 1
  PAI :L2 1
END

TO SUM :N
  IF :N > 20 PR :L2 STOP
```

```
  MAKE "L2 SE :L2 ( PICK :L1 :N ) +
( PICK :L1 :N + 1 ) + ( PICK :L1 :N
+ 2 ) + ( PICK :L1 :N + 3 )
  SUM :N + 1
END

TO PAI :L :K
  IF :K > 20 ( PRINT1 "MAX= :M "' ' )
( PRINT1 PICK :L1 :I "' ' PICK
:L1 :I + 1 "' ' PICK :L1 :I + 2 "' '
PICK :L1 :I + 3 ) ( PR "—— :I :I
+ 1 :I + 2 :I + 3 ) PR "'' ( PRINT
1 "MIN= :T "' ' ) ( PRINT1 PICK :L
1 :J "' ' PICK :L1 :J + 1 " ' ' PICK
 :L1 :J + 2 "' ' PICK :L1 :J + 3 )
( PR "—— :J :J + 1 :J + 2 :J + 3 )
TOPLEVEL
  IF :M < PICK :L :K MAKE "M PICK
:L :K MAKE "I :K
  IF :T > PICK :L :K MAKE "T PICK
:L :K MAKE "J :K
  PAI :L :K + 1
END

?YPZS
19 1 21 7 9 36 4 2 15 31 3 24 11
20 6 37 17 15 10 18 19 1 21
```

48 38 73 56 51 57 52 51 73 69 58

61 74 80 75 79 60 62 48 59

MAX=80 20 6 37 17--- 14 15 16 17

MIN=38 1 21 7 9--- 2 3 4 5

结果说明，最大的和为80，它是由第14～17的四个数20、6、37及17相加的。而最小的和为38，它是由第2、3、4、5四个数1、21、7和9相加的。

13-4 色彩演示

CEC-I LOGO在彩显屏幕上，有丰富多采色泽艳丽的绘画本领。编写一个使用各种颜色不断变化画图的色彩演示程序。

解 采用不同的笔色和底色，在彩显屏幕上可画颜色不同的美丽图案。在单显屏幕上也能改变画线的粗细和灰度的深浅。

法1 建立两个过程，使笔色和底色都从0变到6。相当于嵌套的二重循环。画旋转移位的正方形，主过程BGPC1，子过程BP。程序如下：

程序13-4-1

```
TO BGPC1
  HOME PU FD 25 PD
  MAKE "B 0 MAKE "P 0
  BP :B :P
```

```
END

TO BP :B :P
  IF :B > 6 STOP
  CS BG :B PC :P SPLITSCREEN
  CURSOR 0 21
  ( PRINT1 "BG= :B "' ' "PC= :P )
  REPEAT 12 [REPEAT 4 [FD 70 RT 90]
RT 30]
  REPEAT 500 [MAKE "V 1]
  MAKE "P :P + 1
  IF :P > 6 MAKE "P 0 MAKE "B :B + 1
  BP :B :P
END
```

法 2 建立三个过程，笔色、底色分别各由一个过程 递归完成。见子过程 BGBG、PCPC。主过程 BGPC2 总控。程序如下：

程序 13-4-2

```
TO BGPC2
  SPLITSCREEN .ASPECT .9 BGBG 0
END

TO BGBG :B
  IF :B = 7 STOP
  BG :B PCPC 0
  BGBG :B + 1
```

```
END
TO PCPC :P
  IF :P = 7 STOP
  CURSOR 0 21
  ( PR "BG :B "' ' "PC :P )
  PC :P DRAW
  REPEAT 8 [REPEAT 4 [FD 50 RT 90]
RT 45]
  PCPC :P + 1
END
```

法 3 用重复命令，实现笔色、底色变化。程序一个过程。程序如下：

程序 13-4-3

```
TO BGPC3
  MAKE "B 0
  REPEAT 7 [MAKE "P 0 REPEAT 7 [DRAW
SETY 25 CS BG :B PC :P CURSOR 0 21
(PR "BG= :B "------PC= :P ) REPEAT
8 [REPEAT 4 [RT 90 FD 60] RT 45]
  MAKE "P :P+1] MAKE "B :B+1]
END
```

13-5 按要求打印英文单词

读入一批英文单词，要求(1)打印出这些词中以 E 结尾的单词；(2) 打印出含有字母 E 的单词；(3) 计算出 E 在这

些单词中出现的频率(即 E 的个数与所有字母个数之比);(4)计算这些单词的平均长度。

解 LOGO 语言模块化的程序设计,正好一个一个地处理各项要求。对于(1),我们只要用 WE 过程对输入表中的每个英文单词词尾进行检测,是 E,打印,并删去该词。对余下的各项再进行同样的检测,不是 E,不打印,其余工作同上。直至表空即所有单词都被检测为止。

对于(2),用两个过程完成。过程 HE 负责从输入表中取词,过程 ZE 负责对每个词中字母进行检测。只要遇到一个字母是 E,就打印出该单词并返回。再取下一个单词。如果整个单词中所有字母都检查过而无 E,也返回,取下一个。

对于(3),用 ZC 及 ZES 两过程,设变量 M 记所有单词字母总个数。变量 T 记各个单词中字母 E 的个数,N 记所有单词中含字母 E 的总个数。最后打印频率 P(E)= N/M。

对于(4),及以上子过程的控制及调用,建立主过程 ABC 完成。它将所有的英文单词以表的形式作输入。先打印出该表,再调用子过程 LEN 测出表长,即表中单词的个数 K。然后给变量 M、N、T 均赋初值 0。分别调用 WE、HE、ZC 三个子过程。最后一条命令,待全部过程执行完后,打印出单词平均长度 AVWD=M/K。

程序及运行示例如下:

程序 13-5

```
TO ABC :L
  PR "' PR :L
  MAKE "K LEN :L
```

```
  MAKE "M 0
  MAKE "N 0
  MAKE "T 0
  WE :L
  HE :L
  ZC  :L
  ( PR "AVWD= :M / :K )
END

TO WE :L
  IF :L = [ ] PR "' ' STOP
  TEST "E = LAST FIRST :L
  IFT ( PRINT1 FIRST :L "' ' ) WE BF
:L
  IFF WE BF :L
END

TO HE :L
  IF :L = [ ] PR "' ' STOP
  ZE FIRST :L
  HE BF :L
END

TO ZC :L
  IF :L = [ ] ( PR "P(E)= :N / :M )
 STOP
   MAKE "M :M + LEN FIRST :L
   MAKE "T 0
```

```
  ZES FIRST :L
  MAKE "N :N + :T
  ZC BF :L
END

TO ZE :Z
  IF :Z = "' ' STOP
  TEST ( FIRST :Z ) = "E
  IFT ( PRINT1 FIRST :L "' ' ) STOP
  IFF ZE BF :Z
END

TO ZES :Z
  IF :Z = "' ' STOP
  TEST FIRST :Z = "E
  IFT MAKE "T :T + 1 ZES BF :Z
  IFF ZES BF :Z
END

TO LEN :Z
  IF ANYOF :Z = "' ' :Z = [ ] OP 0
  OP 1 + LEN BF :Z
END

?ABC [ARE GFRE HGFD TYEA HGRE]

ARE GFRE HGFD TYEA HGRE
ARE GFRE HGRE
ARE GFRE TYEA HGRE
```

P(E)= .210526

AVWD= 3.8

示例中输入5个单词，以E结尾的3个，含E的4个，E的频度为0.210526，单词平均长度3.8。

13-6 掷 硬 币

试编一模拟掷硬币程序。

解 这是数理统计中著名的试验。一般将一枚硬币投掷N次，出现国徽一面的次数与出现币值一面的次数相同或接近，各占总次数的一半。即它们的概率都是$\frac{1}{2}$。

下面定义过程ZYB,它的输入为投掷的次数。执行后，打印出总次数及两面各自出现的次数。

程序及运行示例如下：

程序 13-6

```
TO ZYB :K
  ( PR "K= :K )
  MAKE "N 0 MAKE "M 0 MAKE "I 1
  AA: MAKE "X RANDOM 2
  IF :X = 1 MAKE "N :N + 1 ELSE MAKE
"M :M + 1
  IF :I = :K ( PR "HM= :N "ZM= :M )
STOP
  MAKE "I :I + 1 GO "AA
END
```

```
?ZYB 100
K= 100
HM= 50 ZM= 50

?ZYB 300
K= 300
HM= 162 ZM= 138

?ZYB 500
K= 500
HM= 248 ZM= 252

?ZYB 1000
K= 1000
HM= 474 ZM= 526
```

13-7 掷 骰 子

编程模拟掷骰子游戏。让计算机产生1～6的随机数。每按空格键一次，相当掷一次骰子，屏幕上出现一个随机数。当你想结束操作，按下E键之后，屏幕上显示出投掷的总次数及1～6分别出现的次数。

解法1 此题要解决两个问题，如何产生1～6六个随机数，及对每次产生的随机数的辨认、计数和显示。还要设置一个总次记数器。

随机数的产生，用1+RANDOM 6即可解决。总计数变量设为N，每产生一次即加1。然后用连续几个条件判断命令，对随机数加以辨认和计数。产生出一个随机数即刻打印。

程序及运行示例如下：

程序 13-7-1

```
TO ZSZ
  MAKE "N 0
  MAKE "D1 0 MAKE "D2 0
  MAKE "D3 0 MAKE "D4 0
  MAKE "D5 0 MAKE "D6 0
  AA: MAKE "M RC
  IF :M = "E ( PR "TOTAL= :N ) ( PR
"1--- :D1 ) ( PR "2--- :D2 ) ( PR "3---
:D3 ) ( PR "4--- :D4 ) ( PR "5--- :D5)
( PR "6--- :D6 ) STOP
  MAKE "D 1 + RANDOM 6 ( PR "IS :D)
  MAKE "N :N + 1
  IF :D = 1 MAKE "D1 :D1 + 1 GO "AA
  IF :D = 2 MAKE "D2 :D2 + 1 GO "AA
  IF :D = 3 MAKE "D3 :D3 + 1 GO "AA
  IF :D = 4 MAKE "D4 :D4 + 1 GO "AA
  IF :D = 5 MAKE "D5 :D5 + 1 GO "AA
  MAKE "D6 :D6 + 1 GO "AA
END

?ZSZ
IS 2
IS 4
IS 3
```

```
IS 4
IS 3
IS 3
IS 3
IS 2
IS 4
IS 4
IS 2
IS 1
IS 6
IS 4
TOTAL= 14
1--- 1
2--- 3
3--- 4
4--- 5
5--- 0
6--- 1
```

解法 2 不使用单个的变量及若干个条件命令，而使用LOGO复合字，产生数组分别存放1～6的随机数的个数。可以使程序稍加简练。

程序及运行示例如下。键入过程名ZSZ1并回车后，按任意键即可。屏幕不断出现“IS几”字样，即掷的点数。按E键，马上打印出总数及1～6各点出现的次数，并结束运行。

程序及运行结果如下：

程序 13-7-2

```
TO ZSZ1
  MAKE "N 0
  MAKE "I 0
  REPEAT 6 [MAKE "I :I+1 MAKE WORD
"D :I 0]
  AA: MAKE "M RC
  IF :M = "E ( PR "TOTAL= :N ) MAKE
 "I 1 REPEAT 6 [ (PR :I "--- THING
WORD "D :I ) MAKE "I :I+1] STOP
  MAKE "D 1 + RANDOM 6 ( PR "IS :D)
MAKE WORD "D :D ( THING WORD "D :D)
+ 1 MAKE "N :N + 1
  GO "AA
END

?ZSZ1
IS 2
IS 2
IS 3
IS 5
IS 3
IS 4
IS 4
IS 2
IS 3
```

```
IS 3
IS 2
IS 1
IS 6
IS 1
IS 2
IS 6
IS 1
TOTAL= 17
1 --- 3
2 --- 5
3 --- 4
4 --- 2
5 --- 1
6 --- 2
```

13-8 豆子下落实验

如图 13-2 所示的一个容器，入口处有一个通道，经过一个菱形隔板后变成两个通道，再经过两个菱形隔板又变成为三个通道；以此类推，最底层为七个格子。有 N粒豆子从容器的入口处一个一个地落下，每经过一个菱形隔板，豆子向左右两方向下落的机会相等。编一程序计算落入每一个格子中的豆子的数目。

解 这是用计算机模拟概率现象的一个典型题目。统计物理学中专门有这样的实验装置。每一粒豆子从入口进入，

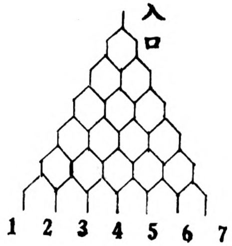


图 13-2

每经过一个菱形隔板，向左或向右下落的可能都有。但可以想到，经过六层菱形隔板，一直向左下落进入1，与一直向右下落进入7的机会都很少。而最多可能的机会是3、4、5。即既有向左，也有向右。如果用随机函数RANDOM 2，使产生的输出值0或1，分别表示向左和向右。当其值为1时，给“数组”变量WORD ″A :I 加1。最后该变量是几，就表示落入第几个格子。这样N粒豆子落下之后，自然就形成一个正态分布的概率现象。

定义DZ过程。I记数。数组″A :I 记格子数。第一、二行清零。第三行有N粒豆子，重复N次，内层嵌套，对每粒豆子重复六次，统计6次可能出现的情况，给数组记数。第四、五行将数组值打印输出。

程序及运行示例如下。可以看出，无论100粒豆子，还是500粒子，它都符合正态分布的规律。两边的机会都较少，中间的机会都较多。

程序 13-8

```
TO DZ :N
  MAKE ″I 0
```

```
  REPEAT 7 [MAKE "I :I+1 MAKE WORD
"A :I 0]
  REPEAT :N [MAKE "I 1 REPEAT 6 [IF
  (RANDOM 2 )=1 MAKE "I :I+1] MAKE
  WORD "A :I (THING WORD "A :I )+1]
  MAKE "I 0
  REPEAT 7 [MAKE "I :I+1 PRINT1 SE
(THING WORD "A :I ) CHAR 32] PR "
END
?DZ 100
3 11 15 30 25 12 4
?DZ 500
6 41 136 159 110 42 6
```

13-9　字符组成移动的飞机

编程在屏幕上产生一个可以围绕屏幕四周飞行的由字符组成的飞机。如图 13-3 所示。

解　字符在不同的行和列，连续变化就产生动感。我们编制飞行的飞机，程序由五个过程构成。主过程 FJ。控制调用四个子过程 SHA、YOU、XIA、ZUO。它们分别产生向上、向右、向下和向左飞行的飞机。这个飞机可以顺时针方向在屏幕周围不断飞行。

程序如下。只要键入 FJ 回车即可。

程序 13-9

```
TO FJ
```

```
  ND
  SHA 18
  YOU 0
  XIA 0
  ZUO 34
  FJ
END

TO SHA :I
  IF :I = 0 STOP
  CLEARTEXT
  CURSOR 2 :I PR "^
  CURSOR 1 :I + 1 PR "^#^
  CURSOR 0 :I + 2 PR "#####
  CURSOR 2 :I + 3 PR "#
  CURSOR 2 :I + 4 PR "#
  CURSOR 1 :I + 5 PR "###
  SHA :I - 1
END

TO YOU :I
  IF :I = 35 STOP
  CLEARTEXT
  CURSOR :I + 3 0 PR "#
  CURSOR :I 1 PR "# CURSOR :I + 3 1
  PR "#>
  CURSOR :I 2 PR "#####>
```

```
  CURSOR :I 3 PR "# CURSOR :I + 3 3
  PR "#>
  CURSOR :I + 3 4 PR "#
  YOU :I + 1
END

TO XIA :I
  IF :I = 18 STOP
  CLEARTEXT
  CURSOR 36 :I PR "###
  CURSOR 37 :I + 1 PR "#
  CURSOR 37 :I + 2 PR "#
  CURSOR 35 :I + 3 PR "#####
  CURSOR 36 :I + 4 PR "V#V
  CURSOR 37 :I + 5 PR "V
  XIA :I + 1
END

TO ZUO :I
  IF :I = 0 STOP
  CLEARTEXT
  CURSOR :I + 2 18 PR "#
  CURSOR :I + 1 19 PR "<# CURSOR :I
  + 5 19 PR "#
  CURSOR :I 20 PR "<#####
  CURSOR :I + 1 21 PR "<# CURSOR :I
  + 5 21 PR "#
```

```
  CURSOR :I + 2 22 PR "#
  ZUO :I - 1
END
```

说明，为了表示“飞机”向不同方向飞行，机前分别用字符“∧”、“>”、“∨”、“<”表示向上、向右、向下及向左为机头。图 13-3 表示向上飞行的飞机。

```
    ∧
  ∧#∧
#####
    #
    #
  ###
```

图　13-3

13-10　打印九九乘法表

编程打印“九九”乘法表。

解法 1　“九九”乘法表中，乘数和被乘数都是从 1 变到 9。建立两个递归过程JU1和 JU2嵌套调用。控制变量分别从1到9，求它们的乘积。为在屏幕上显示全部内容，可以使用光标位置命令。程序如下：

程序 13-10-1

```
TO JU1 :X
  IF :X > 9 STOP
  JU2 1
  JU1 :X + 1
END

TO JU2 :Y
```

```
  IF :Y > 9 STOP
  TEST :X < 6
  IFT CURSOR ( :X - 1 ) * 8 :Y ( PRINT1
WORD :Y "* :X "= ) PR :X * :Y
  IFF CURSOR ( :X - 6 ) * 8 :Y + 10
  ( PRINT1 WORD :Y "* :X "= ) PR :X
* :Y
  JU2 :Y + 1
END
```

解法 2 用重复命令的嵌套，内外各重复 9 次，也能得到"九九"表。不过表的显示是在屏幕上不断滚动的。程序及部分运行结果如下：

程序 13-10-2

```
TO JJB
  MAKE "K 1
  REPEAT 9 [MAKE "J 1 REPEAT 9 [PR
(SE :J "* :K "= :J*:K ) MAKE "J :J+
1] MAKE "K :K+1 PR "']
END

?JJB
1 * 1 = 1
2 * 1 = 2
3 * 1 = 3
4 * 1 = 4
5 * 1 = 5
```

6 * 1 = 6

7 * 1 = 6

8 * 1 = 8

9 * 1 = 9

1 * 2 = 2

2 * 2 = 4

3 * 2 = 6

4 * 2 = 8

5 * 2 = 10

6 * 2 = 12

7 * 2 = 14

8 * 2 = 16

9 * 2 = 18

解法3 在屏幕上不必显示算式及相同的结果。只打印出每个乘积结果的一个斜三角数阵。可用如下的程序。主过程JJB，子过程JJB1。程序及运行结果如下：

程序 13-10-3

```
TO JJB :I ; 1-9
  IF :I > 9 PR "' ' STOP
  JJB1 :I
  JJB :I + 1
END

TO JJB1 :J
  IF :J > 9 PR "'' STOP
  CURSOR :J * 4 - 3 :I + 5
```

```
  PRINT1 :I * :J
  JJB1 :J + 1
END

?JJB 1

  1  2  3  4  5  6  7  8  9
     4  6  8  10 12 14 16 18
        9  12 15 18 21 24 27
           16 20 24 28 32 36
              25 30 35 40 45
                 36 42 48 54
                    49 56 63
                       64 72
                          81
```

13-11　小学生除法演算练习程序

编一个供小学生做两位数除以一位数的除法演算练习程序。要求一开始输入准备作题的个数。学生从键盘回答得数之后给出错(CUO)或对(DUI)的信息。最后评出总得分数。

解　变量A为被除数，用(RANDOM 90)+10 产生两位整数；B 为除数，用(RANDOM 9)+1 产生一位整数。为了保证不出现小数，若 A/B非整数，则返回另外产生。然后由学生回答。并给出信息，再出下一题。程序还用 S记成绩，I记题号。当I大于输入N时，打印出总成绩 S*100/N。

程序及运行示例如下：

程序 13-11

```
TO DIV :N ; (XX/X)
  ND RANDOMIZE MAKE "S 0 MAKE "I 1
  ( PR "N= :N )
  AA: MAKE "A ( RANDOM 90 ) + 10
  MAKE "B ( RANDOM 9 ) + 1
  IF NOT :A / :B = INTEGER :A / :B
GO "AA
  ( PRINT1 :I "> "' ' :A "/ :B "= )
  MAKE "C FIRST RQ
  IF :C = :A / :B PR "' DUI' MAKE
  "S :S + 1 ELSE PR "' CUO'
  MAKE "I :I + 1
  IF :I > :N PRINT1 "'DEFEN ' PR
:S * 100 / :N STOP ELSE GO "AA
END

?DIV 5
N= 5
1> 55/5=11 DUI
2> 56/8=7 DUI
3> 43/1=43 DUI
4> 40/2=20 DUI
5> 63/9=7 DUI
 DEFEN 100.
```

说明：以上程序，运行之后，在屏幕上的运行结果是完

全正确和清楚的。但为了在打印机上能打印出答案数，必须在程序第七行后增加一条命令PRINT1 :C。

13-12　念数程序

编一个“念数”程序。当你输入0～99间任意一个正整数时，计算机就显示出这个数读法的汉语拼音。如果输入的数是小数，或不在0至99之间，它就显示“CUOLE”，要求重新输入。例如输入35，屏幕显示出“SAN SHI WU”。输入0，显示“LING”。

解法1　数中的任何一个非零数字，在十位或个位，读法都是相同的。因而，我们可以定义一个过程，对无论是十位的数字还是个位的数字，判断后输出它的读法。区别在于是十位数，跟着输出SHI。若是个位数，直接结束。这个过程由子过程 DU完成。

再定义主过程 DS。首先排除不符合要求的几种情况。然后，是0直接打印 LING。大于0，则求出十位数字给YS，个位数字给YG。十位是0的，直接调用子过程DU读个位，非零的先调用子过程读十位，跟着打印SHI，再读个位。

程序及运行示例如下：

程序 13-12-1

```
TO DS :X
IF :X < 0 GO "AA
IF :X > 99 GO "AA
IF NOT INTEGER :X = :X GO "AA
IF :X > 0 GO "BB
```

```
  PR "LING STOP
  AA: PR [CUO LE,CONG LAI] STOP
  BB: MAKE "YS QUOTIENT :X 10
  MAKE "YG REMAINDER :X 10
  IF :YS = 0 PRINT1 "" GO "CC
  DU :YS PRINT1 "SHI
  CC: IF :YG = 0 PR "" STOP
  DU :YG PR "" STOP
END

 TO DU :Y
   IF :Y = 1 PRINT1 "YI STOP
   IF :Y = 2 PRINT1 "ER STOP
   IF :Y = 3 PRINT1 "SAN STOP
   IF :Y = 4 PRINT1 "SI STOP
   IF :Y = 5 PRINT1 "WU STOP
   IF :Y = 6 PRINT1 "LIU STOP
   IF :Y = 7 PRINT1 "QI STOP
   IF :Y = 8 PRINT1 "BA STOP
   PRINT1 "JIU STOP
 END

 ?DS 0
 LING

 ?DS 10
 YISHI
```

```
?DS 35
SANSHIWU

?DS -4
CUO LE,CONG LAI

?DS 5.66
CUO LE,CONG LAI
```

解法2 基本思想与1相同，不同的有两点，一是子过程判断数字之后，不用打印命令和暂停命令，而用输出命令OP，更简单一些。二是主过程中除了排除不符合要求数外，对0、对一位数、对两位数分别处理。对两位数中整十（个位为0者）和非整十者又分别处理。

主过程DS1，子过程DU1。

程序及运行示例如下：

程序13-12-2

```
TO DS1 :X
  IF ( ANYOF :X < 0 :X > 99 NOT INT
EGER :X = :X ) PR [CUO LE,CONG LAI]
STOP
  IF :X = 0 PR "LING STOP
  IF ALLOF :X > 0 :X < 10 PR DU1 :X
STOP
  MAKE "YS QUOTIENT :X 10
  MAKE "YG REMAINDER :X 10
  PRINT1 DU1 :YS PRINT1 "SHI
  IF :YG = 0 PR "'' STOP
```

```
  PR DU1 :YG
END

TO DU1 :Y
  IF :Y = 1 OP "YI
  IF :Y = 2 OP "ER
  IF :Y = 3 OP "SAN
  IF :Y = 4 OP "SI
  IF :Y = 5 OP "WU
  IF :Y = 6 OP "LIU
  IF :Y = 7 OP "QI
  IF :Y = 8 OP "BA
  OP "JIU
END

?DS1 35
SANSHIWU

?DS1 0
LING

?DS1 8
BA

?DS1 -87
CUO LE,CONG LAI

?DS1 123
CUO LE,CONG LAI
```

```
?DS1 9.77
CUO LE,CONG LAI

?DS1 50
WUSHI
```

解法 3 将数字读法的汉语拼音及要输出的信息"CUOLE,CONGLAI"等，全部放置于表 L中。定义子过程DU2，按指定的数，输出表 L 中的项。这样，主过程 DS2 中增加一条赋值命令，子过程DU2 则更简练。其余判断同 2 不变。

程序及运行示例如下：

程序 13-12-2

```
TO DS2 :X
  MAKE "L [LING YI ER SAN SI  WU LIU
  QI BA JIU SHI CUOLE,CONGLAI]
  IF ( ANYOF :X < 0 :X > 99 NOT :X
  = INTEGER :X ) PR DU2 11 :L STOP
  IF :X = 0 PR DU2 0 :L STOP
  IF ALLOF :X > 0 :X < 10 PR DU2 :X
:L STOP
  MAKE "YS QUOTIENT :X 10
  MAKE "YG REMAINDER :X 10
  PRINT1 DU2 :YS :L PRINT1 "SHI
  IF :YG = 0 PR "' ' STOP ELSE PRINT
1 DU2 :YG :L PR "' '
END

TO DU2 :N :L
```

```
  IF :N = 0 OP FIRST :L
  OP DU2 :N - 1 BF :L
END
```

此题编程方法较多，而且还可以使用不同的原语命令。如求整商、求余数等命令，与取整命令有相同效果。读者可选择应用。

另外，扩充程序功能，把它制成一个简易软件，可供学前儿童或小学生游戏认数、学拼音之用。下面的程序NIAN·SHU、NIAN、SHU三个过程，即是这样的。运行后，在屏幕中央出现所给数的读法。可以连续进行。按E键结束。程序如下：

程序 13-12-4

```
TO NIAN.SHU
  CLEARTEXT CURSOR 7 3
  PR [NI GEI YI GE SHU, WO LAI NIAN.]
  CURSOR 14 5 PR [( 0<SHU<99 )]
  CURSOR 17 7 PRINT1 [SHU=]
  MAKE "S FIRST RQ
  IF ( ANYOF :S < 0 :S > 99 NOT INT
EGER :S = :S ) THEN CURSOR 12 9 PR
[… [BU DUI!] …] REPEAT 300 [MAKE
"C 200] NIAN.SHU STOP
  NIAN :S
  CURSOR 12 15
  PRINT1 [PLAY AGAIN.(Y/N) ] MAKE
```

```
"D RC
  IF :D = "Y THEN NIAN.SHU
  CLEARTEXT
END

TO NIAN :A
  IF BF :A = "'' THEN CURSOR 18 11
SHU :A STOP
  MAKE "X FIRST :A CURSOR 15 11
  IF LAST :A = 0 CURSOR 17 11
  SHU :X PRINT1 "' '
  PRINT1 "SHI IF LAST :A = 0 STOP
  PRINT1 "' '
  MAKE "Y LAST :A SHU :Y
END

TO SHU :B
  MAKE "Z [LING YI ER SAN SI WU LIU
QI BA JIU]
  REPEAT ( FIRST :B ) [MAKE "Z BF :Z]
  PRINT1 FIRST :Z
END
```

13-13 判断闰年

编程判断所给的任意一公元年号是否闰年。若是闰年则打印YES，不是闰年则打印NO。

解　平年和闰年是这样规定的，凡整百的年号，必须是400的整数倍，才是闰年。不是整百的年号，若是4的整数倍，也是闰年。除此之外，则都是平年。

于是我们可利用逻辑运算符ALLOF、ANYOF及NOT等编程。程序及运行示例如下：

程序 13-13

```
TO YER :N
  IF ANYOF ( ANYOF :N < 0 :N = 0 )
INOT :N = INTEGER :N PR "NO! STOP
  IF ANYOF ( ALLOF NOT :N / 100 =
INTEGER :N / 100 :N / 4 = INTEGER :N
  / 4 ) :N / 400 = INTEGER :N / 400
( PR :N "YEAR "YES ) STOP
  ( PR :N "YEAR "NO )
END

?YER 1990
1990 YEAR NO

?YER 1988
1988 YEAR YES
```

13-14　查星期程序

编一个程序，将年、月、日由键盘输入，按回车后，能输出该天是星期几。

解　应用上面的公式：

$$S = n-1+\frac{n-1}{4}-\frac{n-1}{100}+\frac{n-1}{400}+c$$

然后求出 S/7 的余数，即是星期数。若余数为 0，则表示星期日。式中 n 表示公元年号数，c 表示从这一年的元月一日算起到要查找的日期止的天数(包括要查的这一天)。

例如，要想知道 1949 年 10 月 1 日是星期几，计算方法如下：

$$S=(1949-1)+(1949-1)\left(\frac{1}{4}-\frac{1}{100}+\frac{1}{400}\right)$$
$$+(31+28+31+30+31+30+31+31+30+1)$$
$$=2694$$
$$2694\div 7=384\cdots 6$$

即 1949 年 10 月 1 日是星期六。

有了这样的公式，编程并不十分困难。问题是：每逢平年，2 月份是 28 天，每逢闰年，2 月份是 29 天。程序中应加以区别。

判断平年还是闰年，方法同上题。

程序由两个过程组成。子过程 QRS 用来求日数。它有两个输入，月份数及由一年 12 个月份各月天数组成的表。输出指定月份的总天数。主过程 CXQ，带三个输入，即要查的年、月、日数。

程序及运行示例如下：

程序 13-14

```
TO CXQ :N :Y :R
  MAKE "W :N - 1
  MAKE "S :W + ( QUOTIENT :W 4 ) -
```

```
  ( QUOTIENT :W 100 ) + ( QUOTIENT :W
  400 )
  IF :Y = 1 MAKE "S :S + :R GO "AA
  MAKE "C QRS :Y -1 [31 28 31 30 31 30
  31 31 30 31 30 31]
  MAKE "S :S + :R + :C
  IF ( ALLOF REMAINDER :N 4 = 0 NOT
  REMAINDER :N 100 = 0 :Y > 2 ) MAKE
  "S :S + 1 GO "AA
  IF ( ALLOF REMAINDER :N 4 = 0 REM
AINDER :N 100 = 0 REMAINDER :N 400
= 0 :Y > 2 ) MAKE "S :S + 1
  AA: ( PRINT1 :N ". :Y ". :R "'.IS'
   REMAINDER :S 7 ) PR "' '
END

TO QRS :C :L
  IF :C = 1 OP FIRST :L
  OP ( FIRST :L ) + ( QRS :C - 1 BF
  :L )
END

?CXQ 1988 10 1
1988.10.1.IS 6

?CXQ 1990 1 1
1990.1.1.IS 1
```

```
?CXQ 1949 10 1
1949.10.1.IS 6
```

13-15 打印年历程序

编一个打印年历的程序，要求从键盘上输入任意一个公元年号，计算机就能打印输出这一年的年历来。

解 要求给出一个公元年号，就能打印出这一年历，必须解决三个问题。一，必须知道这一年是平年还是闰年。因为平年 2 月份是 28 天，闰年 2 月份是 29 天。二，应当确定该年的元旦是星期几。三，确定打印格式。

对于第一点，我们可以用 13-13 题的方法确定。即对整百的年号，必须是 400 的整倍数，才是闰年。不是整百的年号，只要是 4 的整倍数，也是闰年。此外，都是平年。

对于第二点，我们可以用 13-14 题的方法确定。在这儿，只要用公式

$$W = Y-1+(Y-1)/4-(Y-1)/100+(Y-1)/400+1$$

再把 W 用 7 除取余就行了。

对于第三点，为了使用光标位置命令 CURSOR，引进一个变量 J，初值从 2 开始，实际打印在第三行。用它来控制行。每行各列的位置由星期数 F 乘 5 加 3 确定。当F = 0 时，表示星期日，J + 1，换行重新开始。这样巧妙地解决了 LOGO 位置命令受列与行两个输入同时限制的问题。

注意，使用此程序时，必须事先接通打印机电源。程序将自动执行连机和脱机。

另外，当你键入主过程名 PRTY 并回车后，按屏幕提示“请输入年号”(拼音)，键入年号，回车。屏幕上是显示出一个月份，清屏，又显示出另一个月份，直到全年 12 个月份。打印机在打印纸上却是连续打印出全年各月份。

下面是程序清单及运行的部分结果：

程序 13-15

```
TO PRTY
  PR [QING SHU RU NIAN HAO]
  MAKE "Y FIRST RQ
  MAKE "W 1 + QUOTIENT ( :Y - 1 ) *
  497 400
  MAKE "L [31 28 31 30 31 30 31 31
30 31 30 31]
  OUTDEV 1
  ( PR "'                         ' :Y )
  PR [------------------------------
----------] PR "' ' PR "' '
  PRTM 1
  OUTDEV 0
END

TO PRTM :M
  ND
  IF :M > 12 STOP
  ( PR "****************** :M
"****************** )
```

```
  ( PR "' SUN' "' MON' "' TUE' "'
WED' "' THU' "' FRI' "' SAT' )
  MAKE "D PICK :M :L
  IF ALLOF :M = 2 ( ANYOF ( ALLOF N
OT REMAINDER :Y 100 = 0 REMAINDER :
Y 4 = 0 ) REMAINDER :Y 400 = 0 ) MAKE
"D :D + 1
  MAKE "J 2
  PRTD 1 :D PR "'' PR "''
  PRTM :M + 1
END

TO PRTD :I :D
  IF :I > :D STOP
  MAKE "F REMAINDER :W 7
  IF :F = 0 MAKE "J :J + 1 PR "''
  CURSOR 5 * :F + 3 :J
  PRINT1 :I
  MAKE "W :W + 1
  PRTD :I + 1 :D
END

TO PICK :I :L
  IF :I = 1 OP FIRST :L
  OP PICK :I - 1 BF :L
END

  ?PRTY
```

```
QING SHU RU NIAN HAO
                    1991
----------------------------------------
*****************  1  *****************
SUN MON TUE WED THU FRI SAT
             1   2   3   4   5
 6   7   8   9  10  11  12
13  14  15  16  17  18  19
20  21  22  23  24  25  26
27  28  29  30  31
*****************  2  *****************
SUN MON TUE WED THU FRI SAT
                         1   2
 3   4   5   6   7   8   9
10  11  12  13  14  15  16
17  18  19  20  21  22  23
24  25  26  27  28
*****************  3  *****************
SUN MON TUE WED THU FRI SAT
                         1   2
 3   4   5   6   7   8   9
10  11  12  13  14  15  16
17  18  19  20  21  22  23
24  25  26  27
STOPPED!, IN LINE
  IF :F = 0 MAKE "J :J + 1 PR ""
```

AT LEVEL 33 OF PRTD。

程序由四个过程构成。PRTY为主过程。PRTM、PRTD及PICK为三个子过程。其中主过程要求输入年号，将全年12个月各月的天数置于表中，赋给变量L。开启打印机并调用子过程PRTM，子过程PRTM控制月份。并判断是否闰年，调整2月份的天数。子过程PRTD控制每月的天数，按其每天的星期数打印出该天的位置。子过程PICK输出表的指定项，供PRTM调用。依次把每个月的天数赋给变量D。

13-16 趣味清屏程序

LOGO文本状态的趣法清屏。

在LOGO中，我们可以使用原语命令CLEA RTEXT清屏，也可以调用系统提供的机器语言子程序清屏(如从光标清至当前行尾入口地址7580，从光标清至屏幕终点入口地址7416，与上述原语命令等效，光标回左上角的入口地址7406等)。

但是，仅仅使用这些命令清屏，总显得单调和呆板，我们可以编制程序而使清屏方式方法多样有趣，生动灵活。

下面就是一个演示程序。屏幕上首先布满R字符，你可以选择0～9中的任一键，机器就表演一种清屏方式，并奏出一段音乐，然后又布满字符，供你选用另一种方式清屏。如要停止运行，只需按一下E键。

整个程序共由16个过程构成。其中CL是主过程。子过程WR显示满屏字符，PK输出表中指定元素，SU演

奏音乐，C0—C9为不同的清屏方式。这10种清屏方式分别是：

0. 左右两边同时从上向下一列一列清除。

1. 从上向下迅速清除。

2. 从左向右一行一行清除。

3. 从中间同时向上下清除。

4. 从中间同时向左右清除。

5. 从上下同时向中间清除。

6. 从下向上迅速清除。

7. 先从右边从上向下，再从左边从下向上各占半幅屏幕清除。

8. 上半屏幕从右向左，下半屏幕从左向右清除。

9. 从四周向中心收缩清除。

这些清除屏幕方式的各个过程，均可单独独立使用，但必须注意它们各自的输入数值的起始及范围，否则将会使系统崩溃。

另外，在实际运行中，只要稍加修改过程中的一些命令，还可产生许许多多新颖有趣的清屏方式，绝不仅限于此10种方式。愿读者自己修改变化。

清屏程序如下：

程序 13-16

```
TO CL
  WR
  MAKE "K RC
  TEST ALLOF ASCII :K > 47 ASCII :K
  < 58
```

```
  IFF .CALL 7406 0 STOP
  MAKE "N P :K + 1 [[C0 0] [C1 0] [C2 0]
[C3 1] [C4 1] [C5 0] [C6 0] [C7 0]
[C8 39] [C9 1]]
  RUN :N SU 1 CL
END

TO WR
  REPEAT 23 [REPEAT 40 [PRINT1 "R]]
END

TO PK :N :L
  IF :N = 1 OP FIRST :L
  OP PK :N - 1 BF :L
END

TO SU :N ; 0-6
  IF :N > 6 STOP
  .DEPOSIT 236 30 * :N
  .CALL 53497 0
  .DEPOSIT 237 20 * :N
  .CALL 53497 0
  SU :N + 1
END

TO C0 :X ; 0
  TEST :X > 19
  IFT CURSOR 0 0 STOP
```

```
  C01 :X 0
  C0 :X + 1
END

TO C01 :X :Y ; 0 0
  IF :Y > 23 STOP
  CURSOR :X :Y
  PRINT1 CHAR 32
  CURSOR 39 - :X :Y
  PRINT1 CHAR 32
  C01 :X :Y + 1
END

TO C1 :N ; 0
  IF :N > 23 ND STOP
  CURSOR 0 :N
  PR CHAR 32
  C1 :N + 1
END

TO C2 :N ; 0
  IF :N > 23 ND STOP
  CURSOR 0 :N
  REPEAT 40 [PRINT1 CHAR 32]
  C2 :N + 1
END

TO C3 :N ; 1
```

```
  IF :N > 12 ND STOP
  CURSOR 0 12 - :N
  .CALL 7580 0
  CURSOR 0 11 + :N
  .CALL 7580 0
  C3 :N + 1
END

TO C4 :N ; 1
  IF :N > 20 ND STOP
  .DEPOSIT 32 20 - :N
  .DEPOSIT 33 2 * :N
  .CALL 7406 0
  C4 :N + 1
END

TO C5 :N ; 0
  IF :N > 11 CURSOR 0 0 STOP
  CURSOR 0 :N
  .CALL 7580 0
  CURSOR 0 23 - :N
  .CALL 7580 0
  C5 :N + 1
END

TO C6 :N ; 0
  IF :N > 23 STOP
```

```
  CURSOR 0 23 - :N
  .CALL 7580 0
  C6 :N + 1
END

TO C7 :N ; 0
  IF :N > 23 C6 0 STOP
  CURSOR 20 :N
  .CALL 7580 0
  C7 :N + 1
END

TO C8 :N ; 39
  IF :N < 0 .DEPOSIT 35 24 C81 1 STOP
  .DEPOSIT 35 12
  .DEPOSIT 32 :N
  .DEPOSIT 33 40 - :N
  .CALL 7406 0
  C8 :N - 1
END

TO C81 :X ; 1
  IF :X > 40 STOP
  .DEPOSIT 33 :X
  .CALL 7406 0
  C81 :X + 1
END
```

```
TO C9 :N ; 1
  IF :N >12 CURSOR 0 0 STOP
  .DEPOSIT 35 :N .CALL 7406 0
  .DEPOSIT 34 24 - :N .CALL 7406 0
  .DEPOSIT 35 24 .DEPOSIT 34 0
  .DEPOSIT 33 :N
  .DEPOSIT 32 :N - 1 .CALL 7406 0
  .DEPOSIT 32 ( 40 - ( 2 * :N -1 ) )
  .CALL 7406 0
  .DEPOSIT 32 0 .DEPOSIT 33 40
  C9 :N + 1
END
```

附录 软件制作

一 《LOGO键控绘图》软件简介

(一) 软件功能

使用本软件可以编辑、存储、修改及打印各种图形。用户根据使用说明，只要直接使用计算机键盘上几个按键，操纵画笔上、下、左、右移动，从而绘画出各式各样美丽的图形和图案。直观简便，免于编程，更容易发挥想象和创造。

(二) 硬件配置

中华学习机，苹果 Ⅱ 或其兼容机，64 K RAM，一台驱动器。苹果Ⅱ用 LOGO 系统盘(MIT 版)一张。

(三) 操作使用

开机调入绘图系统，执行主过程后，根据屏幕提示选择各项功能。

(1) 输出使用说明；

(2) 输出一组图形；

(3) 进入绘图状态；

(4) 结束退出。

在绘图状态下，可用 U、N、H、J、Y、I、B、M 共八个键，使画笔产生上、下、左、右及左上、右上、左下及右下八个方向移动。0、1、2、3、4、5、6七个键控制笔色。另外，按 A 键同色绘图(即擦除图形)，按 S 键，并给出两个坐标值，则画笔直接移动。

按 CTRL-H 键隐龟

按 CTRL-X 键显龟

按 CTRL-D 键清屏

按 CTRL-A 键选择底色

按 CTRL-C 键图形存盘

按 CTRL-P 键打印图形，还可根据屏幕提示，确定打印方式及密度。

按 CTRL-Q 键可调出磁盘图形进行修改。

绘图过程，可随时阅读使用说明。只需按“？”键。

按CTRL-O键输入边数和边长，可绘制正多边形。

按 ESC 键返回主菜单。

（四）软件主要程序

（1）菜单文件

程序共 8 个过程。清单如下：

```
TO CAIDAN
TO C.1
TO C.2
TO C.3
TO R :A :B
TO C :A :B
TO CT
TO C.4
TO CAIDAN
  C.1 C.2 C.3 C.4
END

TO C.1
```

```
CT
C 9 3 PR [** JIAN KONG HUI TU **]
C 0 5 R 40 "-
C 10 8 PR "<1>…KANSHUOMING
C 10 10 PR "<2>…GUANKANTUXING
C 10 12 PR "<3>…JINRUHUITU
C 10 14 PR "<4>…JIESHU
C 2 17 PR "QINGXUANZE (1-4):
DOS [BSAVE C-1, A$400, L$3F8]
END

TO C.2
CT C 11 1 PR "' ** [SHUO MING] **'
C 16 3 PR "'Y  U  I' C 17 4 PR "'\  !  /'
C 18 5 PR " '\ ! /' C 16 6 PR "'H—  --J'
C 18 7 PR " '/ ! \' C 17 8 PR "'/  !  \'
C 16 9 PR " 'B  N  M' C 9 11
PR " ' {KONG ZHI FANG XIANG]' C 6 13
PR " ' (0) (1) (2) (3) (4) (5) (6)'
C 11 15 PR "'{KONG ZHI BI SE]'
C 1 17 PR "'CTRL-A:DI SE'
C 20 17 PR "'CTRL-P:DA YIN'
C 1 18 PR "'CTRL-D:QING PING'
C 20 18 PR "'CTRL-H:YIN GUI'
C 1 19 PR "'CTRL-X:XIAN GUI'
C 20 19 PR "'CTRL-O:DUO BIAN XING'
```

```
  C 1 20 PR "'CTRL-Q:QU TU'
  C 20 20 PR "'CTRL-C:CUN TU'
  C 1 21 PR "'A:TONG SE ZUO TU'
  C 20 21 PR "'S:YI ZUO BIAO'
  C 1 22 PR "'?:SHUO MING'
  C 20 22 PR "'{ESC}:FAN HUI'
  DOS [BSAVE C-2,A$400,L$3F8]
END

TO C.3
  CT
  C9 4 PR "'** DA YIN TU XING **'
  C0 5 R 40 "=
  C1 7 PR [FEN 4 ZHONG FANG SHI]
  C3 8 PR "'{1}.ZHENG CHANG FANGSHI'
  C3 9 PR "'{2}.FAN XIANG FANG SHI'
  C3 10 PR "'{3}.ZHENG CHANG FANG DA'
  C3 11 PR "'{4}.FAN XIANG FANG DA'
  C3 12 PR "'{0}.FAN HUI'
  C0 13 PR "'XUAN ZE FANG SHI (1-4):'
  C1 15 PR "'FEN 2 ZHONG MI DU'
  C3 16 PR "'{1}.BIAO ZHUN MI DU'
  C3 17 PR "'{2}.BEI MI DU'
  C3 18 PR "'{0}.FAN HUI'
  C0 19 PR "'XUAN ZE MI DU(1-2):'
  DOS [BSAVE C-3,A$400,L$3F8]
```

```
END

TO R :A :B
  REPEAT :A [PRINT1 :B]
END

TO C :A :B
  CURSOR :A :B
END

TO CT
  CLEARTEXT
END

TO C.4
  CT C 10 3 PR "' ** XUAN ZE DI SE
   **'
  C 0 5 R 40 "- C 16 7 PR " (0---6)
  C 13 9 PR [0----<HEI SE>]
  C 13 10 PR [1----<BAI SE>]
  C 13 11 PR [2----<LU SE>]
  C 13 12 PR [3----<HONG SE>]
  C 13 13 PR [4----<HUANG SE>]
  C 13 14 PR [5----<LAN SE>]
  C 13 15 PR [6----<HEI SE> ]
  C 6 17 PR [QING XUAN ZE: ]
  DOS [BSAVE C-4,A$400,L$3F8]
END
```

(2) 图形文件程序

主要程序。程序共 24 个过程，清单如下：

```
TO JI
TO CH :WX :WY
TO .S
TO .BG
TO SP
TO DB
TO DY
TO S
TO RP
TO SM
TO .RQ :A :B :C
TO C :L :H
TO KZ
TO YD
TO XY
TO TU2
TO P1 :L
TO XZ
TO YS
TO TU
TO F
TO M :AA :BB
TO CT
TO DT :T
```

```
TO JI
  IF RC? OP RC
  OP "ABC
END

TO CH :WX :WY
  MAKE "VX :WX MAKE "VY :WY YD
END

TO .S
  A: S .RQ 2 20 "〈X〉 (ZUOBIAO): M "V :O
  IF ( ANYOF :V > 139 :V < - 140 ) GO"A
  B: .RQ 22 20 "〈Y〉(ZUOBIAO): IF (ANYOF
:O > 119 :O < -120 ) GO "B
  C 17 21 P1 "(U/D): M "Q RC C 17 22 P1
"(Y/N): M "L RC
  IF :L = "N F STOP
  IF :Q = "U PU
  F SETXY :V ( :O ) PD
END

TO .BG
  CT DOS [BLOAD C-4]
  D: C 19 17 M "D RC IF ( ALLOF ASCII
:D>47 ASCII :D < 55 ) F BG :D STOP
  PR CHAR 7 GO "D
END
```

```
TO SP
  CT C 14 6 PR [** CUN TU **] C 11 8
  PR "-----------------
  .RQ 12 11 [WENJIANMING: ] C 16 13 P1
  "(Y/N): M "S RC
  IF :S = "Y F HT SAVEPICT :O ST
END

TO DB
  S .RQ 2 20 "'BIAN SHU:' M "V :O .RQ
  25 20 "'BU CHANG:' C 17 21 P1 "(U/D):
  M "Q RC C 17 22 P1 "(Y/N): M "L RC IF
  :L = "N F STOP
  IF :Q = "U PU
  F REPEAT :V [FD :O RT 360/:V] PD
END

TO DY
  CT DOS [BLOAD C-3]
  C: C 22 13 M "A RC IF :A = "1 M "P 1
  GO "A
  IF :A = "2 M "P 33 GO "A
  IF :A = "3 M "P 65 GO "A
  IF :A = "4 M "P 97 GO "A
  IF :A = "0 STOP
  GO "C
  A: C 19 19 M "B RC IF :B = "1 M "D
```

```
  75 GO "B
  IF :B = "2 M "D 76 GO "B
  IF :B = "0 STOP
  GO "A
  B: F YS HT OUTDEV 1 PR [] .DEPOSIT
  1913 :P .DEPOSIT 1145 :D PR CHAR 17
  OUTDEV 0 ST
END

TO S
  CT SPLITSCREEN
END

TO RP
  CT C 14 6 PR [** QU TU **] C 11 8 PR
   "-------------------
   .RQ 12 11 [WENJIANMING: ] C 16 13 P1
   "(Y/N): M "S RC IF :S = "Y F
    READPICT :O
END

TO SM
  CT DOS [BLOAD C-2] C 1 23 M "K RC
END

TO .RQ :A :B :C
  A: C :A :B P1 :C M "O PQ IF :O = []
  GO "A
```

```
  M "O FIRST :O
END

TO C :L :H
  CURSOR :L :H
END

TO KZ
  A: M "K JI IF :K = "ABC YD GO "A
  IF :K = "U SETH 0 CH 0 1 GO "A
  IF :K = "I SETH 45 CH 1 1 GO "A
  IF :K = "J SETH 90 CH 1 0 GO "A
  IF :K = "M SETH 135 CH 1 0 - 1 GO "A
  IF :K = "N SETH 180 CH 0 0 - 1 GO "A
  IF :K = "B SETH 225 CH - 1 0 - 1 GO "A
  IF :K = "H SETH 270 CH - 1 0 GO "A
  IF :K = "Y SETH 315 CH - 1 1 GO "A
  IF :K = "A PC LAST BL TS XY GO "A
  IF :K = "? SM F XY GO "A
  IF :K = "S .S XY GO "A
  M "T ASCII :K IF :T = 1 .BG XY GO "A
  IF :T = 3 SP F XY GO "A
  IF :T = 4 DRAW TU2 XY GO "A
  IF :T = 8 HT XY GO "A
  IF :T = 15 DB XY GO "A
  IF :T = 16 DY F XY GO "A
  IF :T = 17 RP F XY GO "A
```

```
  IF :T = 24 ST XY GO "A
  IF :T = 27 XZ TU2
  IF ( ALLOF :T > 47 :T < 55 ) PC :T
    - 48 GO "A
  PR CHAR 7 XY GO "A
END

TO YD
  SETXY ( XCOR + :VX ) ( YCOR + :VY )
END

TO XY
  MAKE "VX 0 MAKE "VY 0
END

TO TU2
  F HT CS PU SETXY - 139 ( - 95 )
  SETH 0 PD PC 1 REPEAT 2[FD 190 RT 90
  FD 278 RT 90] ST PU HOME PD XY YD KZ
END

TO P1 :L
  PRINT1 :L
END

TO XZ
  A: CT DOS [BLOAD C-1]
  B: C 18 17 M "B RQ IF :B = [ ] GO "B
  M "B FIRST :B IF :B = "1 SM GO "A
```

```
  IF :B = "2 DT 1 GO "A
  IF :B = "3 STOP
  IF :B = "4 GOODBYE
  GO "B
END

TO YS
  REPEAT 200 [MAKE "I 0]
END

TO TU
  .DEPOSIT 233 255 ND XZ ND .ASPECT 1
  P1 [HUI TU YAO FANG KUANG MA ?
(Y/N) :]
  M "Y RC IF :Y = "Y TU2
  DEFINE "TU2 [[] [F HT PU HOME PD
CS ST XY YD KZ]] TU2
END

TO F
  FULLSCREEN
END

TO M :AA :BB
  MAKE :AA :BB
END

TO CT
  CLEARTEXT TEXTSCREEN
```

```
END
TO DT :T
  A: CT M "F SE ( WORD "BLOAD "TU
:T ".PICT ) " DOS :F .DEPOSIT 49234 0
   .DEPOSIT 49232 0 M "K RC .DEPOSIT
49233 0 IF :T = 5 STOP
  IF :K = CHAR 27 STOP ELSE M "T
:T + 1 GO "A
END
```

（五）软件输入方法

启动LOGO，键入菜单文件程序，执行 CAIDAN过程（盘上将自动生成 C-1～C-4 四个文件）。然后，清除内存。任选一些图形，分别以 TU1、TU2……为文件名存盘。再输入图形文件程序，以最后一幅图形的序号作 DT 过程的终止条件（此处为 5，见该过程第 4 行），存盘。应用时，调出图形文件，执行 TU 过程即可。

二 《LOGO 音乐制作》软件简介

（一）软件功能

使用本软件可以方便地创作、存取与续写歌曲曲谱，并由计算机演奏。

（二）硬件配置

同前。

（三）操作使用

调入该软件程序后，键入 YIN 回 车，则显 示软件主菜

单。根据菜单，可选择如下功能：

（1）列磁盘目录；

（2）选择歌曲演奏速度(分五类)；

（3）选择歌曲音调(分十种)；

（4）创作歌曲，选音调高低及节拍长短，按 ESC 键退出，按任一键继续；

（5）续写歌曲，同(4)；

（6）演奏，计算机根据音调节拍演奏；

（7）存盘，输入文件名，机器自动加后缀 .MUSIC；

（8）从磁盘读取，同(7)；

（9）退出。

（四）软件主要程序

（1）菜单文件程序

程序共 8 个过程。清单如下：

```
TO CAIDAN
TO C.1
TO C.2
TO C.3
TO C.4
TO CT
TO C :A :B
TO R :A :B
TO CAIDAN
  C.1 C.2 C.3 C.4
END
TO C.1
```

```
  CT C 9 3 PR "' * *YIN YUE ZHI ZUO * *'
  C 0 5 R 40 "=
  C 9 7 PR "'<1>. LIE MU LU'
  C 9 8 PR "'<2>. XUAN ZE YIN DIAO'
  C 9 9 PR "'<3>. XUAN ZE SU DU'
  C 9 10 PR "'<4>. YIN YUE ZHI ZUO'
  C 9 11 PR "'<5>. XU XIE YIN YUE'
  C 9 12 PR "'<6>. XIU GAI YIN YUE'
  C 9 13 PR "'<7>. YAN CHANG'
  C 9 14 PR "'<8>. CHU CUN YIN YUE'
  C 9 15 PR "'<9>. DIAO QU YIN YUE'
  C 9 16 PR "'<0>. JIE SHU'
  C 2 18 PR [QING XUAN ZE: ]
  DOS [BSAVE C-1, A$400, L$3F8]
END

TO C.2
  CT C 9 3 PR "' * * XUAN ZE YIN DIAO
   * *'
  C 0 5 R 40 "= C 5 8 PR "'
  A---<C> .DIAO␣␣␣␣
  B---<BD>.DIAO' C5 9 PR "'
  C---<D> .DIAO␣␣␣␣
  D---<BE> .DIAO'
  C 5 10 PR "'E---<E> .DIAO␣␣␣␣
   F---<F> .DIAO'
```

```
  C 5 11 PR "'G---⟨BG⟩.DIAO⊔⊔⊔⊔
   H---⟨G⟩ .DIAO'
  C 5 12 PR "'I---⟨BA⟩.DIAO⊔⊔⊔⊔
   J---⟨A⟩ .DIAO'
  C 5 13 PR "'K---⟨BB⟩.DIAO⊔⊔⊔⊔
   L---⟨B⟩ .DIAO'
  C 1 16 PR [QING XUAN ZE: ]
  DOS [BSAVE C-2, A$400, L$3F8]
END

TO C.3
  CT C 11 5 PR "' * XUAN ZE SU DU *'
  C 0 7 R 40 "-
  C 7 9 PR "'1--KUAI SU  2--SHAO KUAI'
  C 7 11 PR "'3--MAN SU  4--SHAO MAN'
  C 7 13 PR "'5--ZHENG CHANG'
  C 2 16 PR [QING XUAN ZE: ]
  DOS [BSAVE C-3, A$400, L$3F8]
END

TO C.4
  CT C 0 3 R 40 "- C 2 5 PR "'YIN FU
   :[1] [2] [3] [4] [5] [6] [7]'
  C 2 7 PR "'YIN DIAO: [DIYIN]
   [ZHONGYIN] [GAOYIN]' C 12 9 PR
   "D--DIYIN
  C 12 10 PR "Z--ZHONGYIN
```

```
  C 12 11 PR "G--GAOYIN C 2 13 PR "'JIE
   PAI :1/8 1/4 1/2 3/4 1 1.5'
  C 12 14 PR "'2 2.5 3 3.5 4 5'
  DOS [BSAVE C-4,A$400,L$3F8]
END

TO CT
  CLEARTEXT
END

TO C :A :B
  CURSOR :A :B
END

TO R :A :B
  REPEAT :A [PRINT1 :B]
END
```

（2）音乐制作主要程序

程序由 26 个过程构成。清单如下：

```
TO LM
TO YD1
TO SD
TO ZZ1
TO XX
TO XG1
TO YC
TO CY
```

```
TO QY
TO YD2
TO ZZ2
TO YY
TO JJ
TO CT
TO XG3
TO XG2
TO R :N :R
TO YIN
TO P1 :A
TO C :A :B
TO XG4
TO YD3 :A
TO E :A
TO D :A :B
TO L :A :B
TO YD4 :A

TO LM
  CT CATALOG PR " P1 "'AN JIAN FAN
  HUI 'L "K RC
END

TO YD1
  CT DOS [BLOAD C-2]
  A: C 14 16 L "F RC IF ( ALLOF ASCII
```

```
:F > 64 ASCII :F < 77 ) YD2 ELSE GO "A
END

TO SD
  CT DOS [BLOAD C-3]
  A: C 15 16 L "S RC IF :S = "1 L "S 40
  STOP
  IF :S = "2 L "S 65 STOP
  IF :S = "3 L "S 132 STOP
  IF :S = "4 L "S 170 STOP
  IF :S = "5 L "S 96 STOP
  GO "A
END

TO ZZ1
  CT DOS [BLOAD C-4] C 14 1 PR "'* ZHI
ZUO *' L "X 1 ZZ2 JJ
END

TO XX
  CT DOS [BLOAD C-4] C 14 1 PR "'* XU
  XIE *' L "X :X + 1 ZZ2 JJ
END

TO XG1
  CT C 11 2 PR [* XIU GAI YIN YUE *]
  C 9 3 R 23 "
  C 5 5 PR "'/Y/ XIU GAI     /ESC/ TUI
```

```
CHU' D 34 7 D 35 20 C 7 17
  P1 "'KAI SHI XU HAO:'
  P: C 22 17 L "C FIRST RQ IF ( ANYOF :C
>:X ( NOT INTEGER :C = :C ) ) GO "P
  A: IF ( :C - 1 ) = :X XG3 STOP
  C 1 19 P1 :C P1 "') ' L "U E ( 8479 +
:C*2 - 1 ) L "D 1 L "G "D MAKE "F 0
  XG2
  IF :D = 0 GO "B
  L "G "Z L "F 0 XG2 IF :D = 0 GO "B
  L "G "G L "F 0 XG2 IF NOT :D = 0 PR"'
  [DIAO BU DUI]' REPEAT 4000 [ ] XG3
  STOP
  B: C 7 19 P1 :F C 11 19 IF :G = "D P1 "'
  DI YIN '
  IF :G = "Z P1 [ZHONG YIN]
  IF :G = "G P1 "'GAO YIN '
  L "J E ( 8479 + :C * 2 ) C 25 19 IF :J =
   6 P1 "1/8
  IF :J = 12 P1 "1/4
  IF :J = 25 P1 "1/2
  IF :J = 37 P1 "3/4
  IF :J = 50 P1 "'1 '
  IF :J = 75 P1 "1.5
  IF :J = 100 P1 "'2 '
  IF :J = 125 P1 "2.5
```

```
  IF :J = 150 P1 "'3 '
  IF :J = 175 P1 "3.5
  IF :J = 200 P1 "'4 '
  IF :J = 225 P1 "4.5
  IF :J = 250 P1 "'5 '
  C 29 19 P1 "(Y/N/ESC) L "C :C + 1 L "K
  RC IF :K = CHAR 27 XG3 STOP
  IF NOT :K ="Y PR [] GO "A
  XG4 GO "A
END

TO YC
  CT D 249 :S C 10 5 PR [[QING TING YIN
  YUE]] .CALL 39387 0 C 11 15
  P1 [[AN JIAN FAN HUI]] L "K RC CT
END

TO CY
  CT C 15 6 PR [CUN YIN YUE]C 8 8 R 24"*
  A: C 10 10 P1 [WENJIANMING: ] L "F RQ
  IF :F = [] GO "A
  L "F ( SE "BSAVE WORD "MUSIC. FIRST
   :F ",A$2120,L INTEGER ( :X * 2 + 3))
  C 17 12 P1 "(Y/N): L "K RC
  IF :K = "Y DOS :F C 13 16 P1 "'QING
  AN JIAN ' L "K RC
END
```

```
TO QY
  CT C 15 6 PR [QU YIN YUE] C 8 8 R 24 "$
  A: C 10 10 P1 [WENJIANMING: ] L "F RQ
  IF :F = [ ] GO "A
  L "F ( SE "BLOAD WORD "MUSIC. FIRST
   :F ) C 17 12 P1 "(Y/N): L "K RC
  IF NOT :K ="Y STOP
  DOS :F L "X ( INTEGER ( E 43616 + 256
  *E 43617 ) /2 ) - 1 C 13 16
  P1 "'QING AN JIAN ' L "K RC
END

TO YD2
  IF :F = "A YD3 8192
  IF :F = "B YD3 8216
  IF :F = "C YD3 8240
  IF :F = "D YD3 8264
  IF :F = "E YD3 8288
  IF :F = "F YD3 8312
  IF :F = "G YD3 8336
  IF :F = "H YD3 8360
  IF :F = "I YD3 8384
  IF :F = "J YD3 8408
  IF :F = "K YD3 8432
  IF :F = "L YD3 8456
END
```

```
TO ZZ2
  Z: C 0 17 P1 :X PR ")
  A: C 6 17 P1 "'[YIN FU (1/2/3/4/5/6/7)] = '
  L "F RC
  IF ( ANYOF ASCII :F < 49 ASCII :F >
  55 ) GO "A ELSE C 31 17 PR :F
  B: C 6 18 P1 "'[YIN DIAO (D/Z/G) ] = ' L
   "G RC C 25 18
  IF :G = "D PR "'[DI YIN]' GO "H
  IF :G = "Z PR "'[ZHONG YIN]' GO "H
  IF :G = "G PR "'[GAO YIN]' GO "H
  GO "B
  H: L "U THING ( WORD :G :F )
  C: C 6 19 P1 "' [JIE PAI] = '
  L "J RQ IF :J = [ ] GO "C
  L "J FIRST :J
  IF :J = "1/8 L "J 6 GO "D
  IF :J = "1/4 L "J 12 GO "D
  IF :J = "1/2 L "J 25 GO "D
  IF :J = "3/4 L "J 37 GO "D
  IF :J = "1 L "J 50 GO "D
  IF :J = "1.5 L "J 75 GO "D
  IF :J = "2 L "J 100 GO "D
  IF :J = "2.5 L "J 125 GO "D
  IF :J = "3 L "J 150 GO "D
  IF :J = "3.5 L "J 175 GO "D
```

```
  IF :J = "4 L "J 200 GO "D
  IF :J = "4.5 L "J 225 GO "D
  IF :J = "5 L "J 250 GO "D
  GO "C
  D: D 8479 + :X * 2 - 1 :U D 8479 + :X
* 2 :J L "K RC IF :K = CHAR 27 STOP
L "X :X + 1 C 0 17 REPEAT 3 [PR "' ␣␣␣
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣
␣␣␣␣␣␣␣␣␣␣␣␣ ']
  GO "Z
END

TO YY
  B: CT DOS [BLOAD C-1]
  A: C 15 18 L "K RC
  IF :K = "1 LM GO "B
  IF :K = "2 YD1 GO "B
  IF :K = "3 SD GO "B
  IF :K = "4 ZZ1 GO "B
  IF :K = "5 XX GO "B
  IF :K = "6 XG1 GO "B
  IF :K = "7 YC GO "B
  IF :K = "8 CY GO "B
  IF :K = "9 QY GO "B
  IF :K = "0 GOODBYE
  GO "A
END
```

```
TO JJ
  D 8479 + ( :X + 1 ) * 2 - 1 0 D 8479 +
   ( :X + 1 ) * 2 0 CT
END

TO CT
  CLEARTEXT
END

TO XG3
  D 34 0 D 35 24 CT
END

TO XG2
  C: L "F :F + 1
  IF :U = THING ( WORD :G :F ) MAKE
   "D 0 STOP
  IF :F = 7 STOP
  GO "C
END

TO R :N :R
  REPEAT :N [P1 :R]
END

TO YIN
  ND D 49328 16 D 233 255 DOS [BLOAD B
   .MUSIC] L "X 0 L "S 96 D 39388 32
```

```
  D 39393 33 DOS [BLOAD DIAO] YD3 8192
  JJ YY
END

TO P1 :A
  PRINT1 :A
END

TO C :A :B
  CURSOR :A :B
END

TO XG4
  L "C :C - 1 D 35 24 C 1 22 P1 :C P1 ")
  E: C 7 22 L "F RC
  IF (ANYOF ASCII :F < 49 ASCII :F > 55)
   GO "E ELSE C 7 22 P1 :F
  F: C 11 22 L "G RC IF :G = "D PR [DI
  YIN] GO "G
  IF :G = "Z PR [ZHONG YIN] GO "G
  IF :G = "G PR [GAO YIN] GO "G
  GO "F
  G: L "U THING ( WORD :G :F )
  H: C 25 22 L "J RQ IF :J = [ ] GO "H
  L "J FIRST :J
  IF :J = "1/8 L "J 6 GO "I
  IF :J = "1/4 L "J 12 GO "I
```

```
  IF :J = "1/2 L "J 25 GO "I
  IF :J = "3/4 L "J 37 GO "I
  IF :J = "1 L "J 50 GO "I
  IF :J = "1.5 L "J 75 GO "I
  IF :J = "2 L "J 100 GO "I
  IF :J = "2.5 L "J 125 GO "I
  IF :J = "3 L "J 150 GO "I
  IF :J = "3.5 L "J 175 GO "I
  IF :J = "4 L "J 200 GO "I
  IF :J = "4.5 L "J 225 GO "I
  IF :J = "5 L "J 250 GO "I
  GO "H
  I: D 8479 + :C * 2 - 1 :U
  D 8479 + :C * 2 :J
  C 1 22 R 38 "' ' D 35 20
END

TO YD3 :A
  L "F :A YD4 "D YD4 "Z YD4 "G .GCOLL
END

TO E :A
  OP .EXAMINE :A
END

TO D :A :B
```

```
.DEPOSIT :A :B
END

TO L :A :B
  MAKE :A :B
END

TO YD4 :A
  L "D 1
  A: MAKE WORD :A :D E :F
  MAKE "F :F + 1 IF :D = 7 STOP
  MAME "D :D + 1 GO "A
END
```

(3) 机器语言程序

```
*2000. 211F
2000-  FF  FF  F0  E2  CA  B4  A0  97
2008-  87  78  71  65  5A  50  4C  43
2010-  3C  39  32  2D  28  00  00  00
2018-  FF  FE  E2  D6  BE  AA  97  8F
2020-  7F  71  6B  5F  55  4C  47  3F
2028-  39  35  30  2A  26  00  00  00
2030-  FF  F0  D6  CA  B4  A0  8F  87
2038-  78  6B  65  5A  50  47  43  3C
2040-  35  32  2D  28  25  00  00  00
2048-  FE  E2  CA  BE  AA  97  87  7F
2050-  71  65  5F  55  4C  43  3F  39
2058-  32  30  2A  26  22  00  00  00
2060-  F0  D6  BE  B4  A0  8F  7F  78
2068-  6B  5F  5A  50  47  3F  3C  35
2070-  30  2D  28  25  20  00  00  00
```

2078-	E2	CA	B4	AA	97	87	78	71
2080-	65	5A	55	4C	43	3C	39	32
2088-	2D	2A	26	22	1E	00	00	00
2090-	D6	BE	AA	A0	8F	7F	71	6B
2098-	5F	55	50	47	3F	39	35	30
20A0-	2A	28	25	20	1C	00	00	00
20A8-	CA	B4	A0	97	87	78	6B	65
20B0-	5A	50	4C	43	3C	35	32	2D
20B8-	28	26	22	1E	1B	00	00	00
20C0-	BE	AA	97	8F	7F	71	65	5F
20C8-	55	4C	47	3F	39	32	30	2A
20D0-	26	25	20	1C	19	00	00	00
20D8-	B4	A0	8F	97	78	6B	5F	5A
20E0-	50	47	43	3C	35	30	2D	28
20E8-	25	22	1E	1B	18	00	00	00
20F0-	AA	97	87	7F	71	65	5A	55
20F8-	4C	43	3F	39	32	2D	2A	26
2100-	22	20	1C	19	16	00	00	00
2108-	A0	8F	7F	78	71	65	5A	50
2110-	47	3F	3C	39	32	2D	28	25
2118-	20	1E	1C	19	16	00	00	00

*99A0. 9A7F

99A0-	A5	9E	05	9F	F0	13	A5	9E
99A8-	38	E9	01	85	9E	A5	9F	E9
99B0-	00	85	9F	20	BA	99	4C	A0
99B8-	99	60	AD	30	C0	A5	A0	85
99C0-	FC	A5	A1	85	FD	A5	FC	05
99C8-	FD	F0	EE	A5	FC	38	E9	01
99D0-	85	FC	A5	FD	E9	00	85	FD
99D8-	4C	C5	99	A9	20	8D	81	9A
99E0-	A9	21	8D	82	9A	20	80	9A
99E8-	C9	00	F0	42	48	20	80	9A
99F0-	C9	00	F0	3A	85	FF	A5	F9
99F8-	85	FE	20	2F	9A	A9	00	85

9A00-	FC	85	FD	85	FF	68	48	85
9A08-	FE	20	56	9A	A5	FA	85	9E
9A10-	**A5**	FB	85	9F	68	85	A0	A9
9A18-	00	85	A1	06	9E	26	9F	06
9A20-	9E	26	9F	20	A0	99	A9	20
9A28-	20	A8	FC	4C	E5	99	60	48
9A30-	98	48	8A	48	D8	A5	FF	85
9A38-	FD	A9	00	85	FA	85	FB	A2
9A40-	08	46	FD	90	03	18	65	FE
9A48-	6A	66	FA	CA	D0	F3	85	FB
9A50-	68	AA	68	A8	68	60	48	98
9A58-	48	8A	48	A0	10	06	FA	26
9A60-	FB	26	FC	26	FD	38	A5	FC
9A68-	E5	FE	AA	A5	FD	E5	FF	90
9A70-	06	86	FC	85	FD	E6	FA	88
9A78-	D0	E3	68	AA	68	A8	68	60

（五）软件输入方法

在监控状态输入机器语言程序，分别以DIAO和B.MUSIC为名存盘。

启动LOGO输入菜单文件程序，执行CAIDAN过程，盘上将生成C-1～C-4四个文件，然后清除内存。

输入音乐制作程序，存盘。应用时，调出该程序，执行YIN过程即可。

参考书目及文献

[1] [美]哈罗德·艾贝尔森著，胡扬骅译，《AppleⅡ LOGO程序设计》，电子工业出版社，1987 年 3 月。

[2] [美]丹尼尔·瓦特著，吕学礼 陶振宗 曾宪源 方景萍编译，《计算机程序设计 LOGO 语言》，文化教育出版社，1986 年 3 月。

[3] 俞士汶著，《LOGO语言教程》，北京大学出版社，1987 年 4月。

[4] 熊光魁 倪雪荪 陈静然编著，《LOGO 语 言程序设计》，上海科学技术出版社，1988 年 3 月。

[5] 陈树楷 潘懋德 刘雨著，《怎样辅导青少年学习计算机程序设计》，电子工业出版社，1985 年 9 月。

[6] 吴文虎 范俊堂编著，《中学计算机教程习题与解答》，清华大学出版社，1987 年 8 月。

[7] 禹鹏 刘胜利 李全瑞著，《BASIC 语言程序设计例题选》，北京科学技术出版社，1984年 10 月。

[8] 中国计算机软件与技术服务总公司 成都分公司 成都电子研究所合办，《软件报》，1988～1989 年。

[9] 中国福利会，《学生计算机世界》报，1988～1989 年。

0517 8834

责任编辑　贺品品
技术设计　张建飞
封面设计　吕　佳

ISBN 7-5613-0487-1
G·357　定价：4.50元

CEC—I LOGO编程百例

陕西师范大学出版社